A PREDICTIVE MODEL FOR HEPATITIS USING AN ENSEMBLE FEATURE SELECTION TECHNIQUE AND ANFIS

BY

GODSFAVOUR OHIKUAREME AJOGBOR

PSC1805786

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF COMPUTING,

UNIVERSITY OF BENIN,

BENIN CITY,

EDO STATE, NIGERIA.

MAY, 2024.

ATTESTATION

I, GODSFAVOUR OHIKUAREME AJOGBOR hereby attest to the importance and significance of the research topic “A PREDICTIVE MODEL FOR HEPATITIS USING AN ENSEMBLE FEATURE SELECTION TECHNIQUE AND ANFIS"

_____ _____

GODSFAVOUR OHIKUAREME AJOGBOR DATE

(Student)

APPROVAL

This project is hereby approved by the department of computer science in partial fulfilment of the requirement for the award of Bachelor of Science Degree (B.Sc.) in Computer Science of the University of Benin, Benin City, Nigeria.

_____ _____

DR. E. C. IGODAN DATE

(Project Supervisor)

_____ _____

DR. A.R. USIOBAIFO DATE

(Head of Department)

CERTIFICATION

This is to certify that this project work was carried out by GODSFAVOUR OHIKUAREME AJOGBOR with matriculation number PSC1805786 in the Department of Computer Sciences, University of Benin, and it is adequate in scope and content for the award of Bachelor Science Degree in Computer Science of the University of Benin.

_____ _____

DR. E. C. IGODAN DATE

(Project Supervisor)

DEDICATION

This project is dedicated to God Almighty, to my dad and mom, close friends, and supportive associates including my Course mates, expressing gratitude for their unwavering love, encouragement, and invaluable contributions throughout my academic journey.

ACKNOWLEDGEMENTS

My profound gratitude is to God Almighty for the knowledge, wisdom and understanding that he bestowed on me to carry out this project.

To my project supervisor, Dr. E.C. Igodan, for your incredible patience, motivation, support and guidance throughout my project process at the University of Benin, I am thankful and sincerely grateful. Special gratitude goes to the Head of Department of Computer Science, Prof. Godspower Okuobase, my project coordinator Dr. E.C. Igodan. Also, I want to show my profound appreciation to my highly esteemed lecturers in the Department of Computer Science, Prof.A.Imiavan, Mr. S.O.P. Oliomogbe, Dr. F.O. Chete, Prof. F.I Amadin, Prof. K.C. Ukaoha, Dr. Mrs. A.R. Usiobaifo, F.O. Oliha, PhD, Dr. (Mrs.) R.O. Osaseri, Dr. (Mrs.) Aziken, Mr. Idehen for their dedicated tutoring and mentoring. You have all worked hard to set me on the proper road in my professional pursuit and to implant in me sound knowledge about important parts of life. God bless you all.

This project would not have been possible without them who supported me morally and financially. May God Almighty continue to bless each and every one of you.

CONTENTS

ATTESTATION I
APPROVAL II
CERTIFICATION III
DEDICATION IV
ACKNOWLEDGEMENTS V
CONTENTS VI
LIST OF TABLES IX
LIST OF FIGURES 10
ABSTRACT 11
CHAPTER 1 12
INTRODUCTION 12
1.1 Background of the Study 12
1.2 Statement of the Problem 13
1.3 Aim and Objectives of this Study 13
1.4 Methodology 13
1.5 Scope of the Study 14
1.6 Significance of Study 14
CHAPTER 2 15
LITERATURE REVIEW 15
2.1 Overview 15
2.1.1 Hepatitis 15
2.1.2 Mode of Treatment 16
2.2 Artificial Intellegence 17
2.2.1 ANFIS 18
2.2.2. Ensemble Method 19
2.2.3. Different Methods 20
2.3 Feature Selection Method 21
2.3.1 Filter methods 21
2.3.1.1 Merits/Demerits 21
2.3.2 Wrapper methods 22
2.3.2.1 Merits/Demerits 22

2.3.3 Embedded methods 23
2.3.3.1 Merits/Demerits 23
2.4 ML Algorithm 24
2.4.1 Support Vector Machine 25
2.4.2 Supervised Learning 25
CHAPTER 3 35
RESEARCH METHODOLOGY 35
3.1 Data Collection 35
3.1.1 Source Selection: 35
3.1.2 Dataset Description: 35
3.1.3 Data Relevance: 35
3.1.4 Data Integrity: 36
3.1.5 Data Privacy and Ethics: 36
3.2 Data Preprocessing 36
3.2.1 Handling Missing Values: 36
3.2.2 Encoding Categorical Variables: 36
3.2.3 Feature Scaling: 37
3.2.4 Feature Engineering: 37
3.2.5 Data Splitting: 37
3.3 Ensemble Feature Selection 37
3.3.1 ReliefF: 37
3.3.2 Fisher's Scores: 38
3.3.3 Correlation: 38
3.3.4 Ensemble Selection: 39
3.4 Model Development 39
3.4.1 Model Architecture: 39
3.4.2 Model Training: 40
3.4.3 Hyperparameter Tuning: 40
3.3.4 Model Evaluation: 41
3.5 Model Evaluation: 41
3.5.1 Performance Metrics: 41
3.5.2 Cross-Validation: 42
3.5.3 Test Dataset Evaluation: 42

3.5.4 Visualizations: 42
3.5.5 Interpretability: 43
3.5.6 Comparative Analysis: 44
CHAPTER 4 45
IMPLEMENTATION AND RESULTS 45
4.1 System Requirements 45
4.1.1 Hardware Requirements 45
4.1.2 Software Requirements 46
4.2 Tools and Libraries 46
4.2.1 Programming Languages Used 46
4.2.2 Integrated Development Environment (IDE) 46
4.3 Data Preprocessing 47
4.3.1 Handling Missing Values 47
4.3.2 Encoding Categorical Variables 47
4.3.3 Convert Target Variable to Numerical 48
4.3.4 Data Visualization 48
4.4 Ensemble Feature Selection 49
4.4.1 ReliefF, Fisher's Scores, and Correlation 49
4.5 Model Development 51
4.5.1 ANFIS Model 51
4.6 Model Evaluation 53
4.6.1 Performance Metrics 53
CHAPTER 5 56
CONCLUSION 56
5.1 Discussion of Findings 56
5.2 Implications of Research 57
5.3 Future Directions 57
5.4 Conclusion 57
REFERENCE 59

LIST OF TABLES

Table 2- 1: Review of related Literature 27

LIST OF FIGURES

Figure 2-1 Liver Disease and Hepatitis 118
Figure 2 2: Artificial Intelligence (AI) 120
Figure 2- 3 Number of classifiers in error. ..... 2**Error! Bookmark not defined.**
Figure 2- 4: Support Vector Machine 25
Figure 3- 1 Fisher's Score 31
Figure 3- 2 Grid Search 34
Figure 3- 3: ROC curve 36
Figure 3- 4: Sensitivity analysis of ANFIS model for compressional (on left) and shear (on right) wave slowness 37
Figure 4- 1 Handle missing values 40
Figure 4- 2: Encode categorical variables 40
Figure 4- 3: Convert target variable to numerical 41
Figure 4- 4: Distribution of Target Variable 42
Figure 4- 5: Perform feature selection using SelectKBest with ANOVA F-value for ReliefF . **Error! Bookmark not defined.**
Figure 4- 6: Feature Selection Visualization 44
Figure 4- 7: Predict using ANFIS model 45
Figure 4- 8: Plot accuracy over epochs 46
Figure 4- 9: Plot accuracy over epochs……...
…….47
Figure 4-10 Compute performance metrics ……..47
Figure 4-11 Visualize performance metrics comparison ……..……47

ABSTRACT

Hepatitis is a major global health issue, impacting millions worldwide. Early detection and precise diagnosis are essential for effective treatment and preventing complications. This study proposes a predictive model for hepatitis diagnosis using ensemble feature selection techniques and Adaptive Neuro-Fuzzy Inference Systems (ANFIS). The goal is to develop a robust and accurate model to assist healthcare professionals in promptly diagnosing hepatitis cases.

The methodology follows the CRISP-DM framework, including data collection, preprocessing, ensemble feature selection, ANFIS model development, and evaluation using performance metrics like accuracy, precision, recall, and F1-score. Publicly available hepatitis datasets are utilized for experimentation and validation.

The results demonstrate that ensemble feature selection effectively identifies informative hepatitis diagnosis features, enhancing the model's predictive performance. The ANFIS model, trained on the selected features, achieves high accuracy and balanced performance metrics, indicating its effectiveness in accurately diagnosing hepatitis cases.

The implications include early hepatitis detection, healthcare resource optimization, and potential for personalized medicine. Future directions involve integrating additional data sources, continuous model refinement, and clinical validation studies.

This study contributes to advancing predictive modeling for hepatitis diagnosis and highlights the importance of interdisciplinary collaboration between healthcare and data science disciplines in addressing global health challenges.

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

Hepatitis, a significant global health concern, is an inflammation of the liver that can result from various causes, including viral infections, autoimmune diseases, and metabolic disorders (Smith et al., 2018). The World Health Organization estimates that approximately 325 million people worldwide are living with chronic hepatitis, with the majority unaware of their infection (World Health Organization, 2021).

Early detection and accurate diagnosis are crucial for effective treatment and disease management. Traditional diagnostic methods for hepatitis, such as serological tests and liver biopsies, may lack accuracy or require invasive procedures, leading to potential complications and discomfort for patients. Therefore, there is a need for robust predictive models that can accurately classify hepatitis cases based on patient data, thereby assisting healthcare professionals in making informed decisions.

Over the years, the introduction of Artificial Intelligence(AI) has contributed to the development of different models in the prediction and diagnosis of varied diseases. Some of the techniques in the field of AI serving these purposes include machine learning(ML) and ensemble learning methods (Igodan et al., 2022). Machine learning techniques offer promising avenues for developing predictive models to aid in hepatitis diagnosis and prognosis. These techniques can analyze large datasets, identify patterns, and generate predictive models that can assist healthcare providers in diagnosing and managing hepatitis (Smith et al., 2018).

Ensemble feature selection methods are instrumental in improving the performance of predictive models by identifying pertinent features from intricate datasets. Techniques such as ReliefF, Fisher's scores, and Correlation are commonly utilized to extract informative features to enhance model accuracy. ReliefF is a feature selection method that assesses the relevance of features based on their ability to distinguish between instances that are close to each other in the dataset (Kononenko, 1994). Fisher's scores, on the other hand, rank features based on their discriminatory power, with higher scores indicating more relevant features (Duda et al., 2001).

1.2 Statement of the Problem

In light of the strides made in medical technology, the diagnosis of hepatitis continues to present a formidable challenge, primarily owing to the intricate interplay of various factors contributing to the disease's manifestation. Conventional diagnostic techniques, while valuable, often fall short in terms of accuracy and may necessitate invasive procedures, imposing burdensome challenges on patients and healthcare providers alike. Consequently, there exists a compelling demand for the development of robust predictive models. These models, leveraging the wealth of patient data available, have the potential to offer unparalleled precision in classifying hepatitis cases. By harnessing the power of advanced analytics and machine learning algorithms, such models can empower healthcare professionals with actionable insights, facilitating more informed decision-making processes. In doing so, they not only enhance the efficiency and efficacy of hepatitis diagnosis but also pave the way for more personalized and targeted approaches to patient care.

1.3 Aim and Objectives of this Study

The aim of this study is to develop a predictive model for hepatitis using ensemble feature selection techniques and Adaptive Neuro-Fuzzy Inference Systems (ANFIS). The following of this study are to:

Implement ReliefF, Fisher’s scores, and Correlation for feature selection to identify informative features related to hepatitis diagnosis.

1. Develop an ANFIS model trained on the selected features to predict hepatitis cases accurately.
2. Evaluate the performance of the proposed model using appropriate metrics such as accuracy, precision, recall, and F1 score.

1.4 Methodology

The approach of the study is based on CRISP-DM. It contains the following stages:

1. Data Collection: Obtain the hepatitis dataset from Kaggle (source: https://www.kaggle.com/datasets/mragpavank/hepatitis-dataset).
2. Data Preprocessing: Preprocess the collected data to handle missing values, encode categorical variables, standardize numerical features, and ensure data consistency.
3. Ensemble Feature Selection: Apply ReliefF, Fisher's scores, and Correlation techniques to the preprocessed dataset to select the most relevant features for hepatitis prediction.
4. Model Development: Implement an ANFIS model using TensorFlow and train it on the selected features to predict hepatitis cases.
5. Model Evaluation: Assess the performance of the ANFIS model using metrics such as accuracy, precision, recall, and F1 score on a held-out test dataset.

1.5 Scope of the Study

This study focuses on developing a predictive model for hepatitis using ensemble feature selection techniques and ANFIS. It includes feature selection, model development, and evaluation stages to demonstrate the feasibility and effectiveness of the proposed approach. The study will utilize publicly available hepatitis datasets for experimentation and validation.

1.6 Significance of Study

The development of an accurate predictive model for hepatitis holds significant implications for healthcare providers and patients. By leveraging ensemble feature selection techniques and ANFIS, this study aims to improve the accuracy and reliability of hepatitis diagnosis, leading to better patient outcomes and more effective disease management strategies. The findings of this research could contribute to advancements in medical diagnostics and enhance the overall quality of healthcare services provided to individuals affected by hepatitis.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

In this chapter, we delve into a comprehensive review of literature relevant to hepatitis, focusing on its various aspects including etiology, epidemiology, diagnosis, and treatment. Hepatitis is a significant global health issue characterized by inflammation of the liver. It encompasses a spectrum of conditions, predominantly caused by viral infections, autoimmune reactions, or metabolic disorders. This chapter aims to provide a contextual framework for understanding hepatitis and its implications in the medical field.

2.1.1 Hepatitis

Hepatitis, derived from the Greek word "hepar" meaning liver and the Latin suffix "-itis" indicating inflammation, refers to inflammation of the liver tissue. This inflammation can be acute or chronic and is often associated with liver dysfunction. The most common types of hepatitis are hepatitis A, B, C, D, and E, each caused by different viruses. Hepatitis A and E are typically caused by ingestion of contaminated food or water, while hepatitis B, C, and D are transmitted through contact with infected blood or bodily fluids. Autoimmune hepatitis, caused by the immune system attacking liver cells, and alcoholic hepatitis, resulting from excessive alcohol consumption, are examples of non-viral hepatitis. Note Hepatitis E virus (HEV) was discovered during the Soviet occupation of Afghanistan in the 1980s, after an outbreak of unexplained hepatitis at a military camp. A pooled faecal extract from affected soldiers was ingested by a member of the research team. He became sick, and the new virus (named HEV), was detected in his stool by electron microscopy. Subsequently, endemic HEV has been identified in many resource-poor countries. Globally, HEV is the most common cause of acute viral hepatitis. The virus was not initially thought to occur in developed countries, but recent reports have shown this notion to be mistaken. The aim of this Seminar is to describe recent discoveries regarding HEV, and how they have changed our understanding of its effect on human health worldwide (Kamar et al., 2012).

The symptoms of hepatitis can vary depending on the type and severity of the infection but often include fatigue, jaundice, abdominal pain, nausea, and vomiting. While acute hepatitis may resolve on its own, chronic hepatitis can lead to serious complications such as liver cirrhosis, liver failure, and hepatocellular carcinoma.

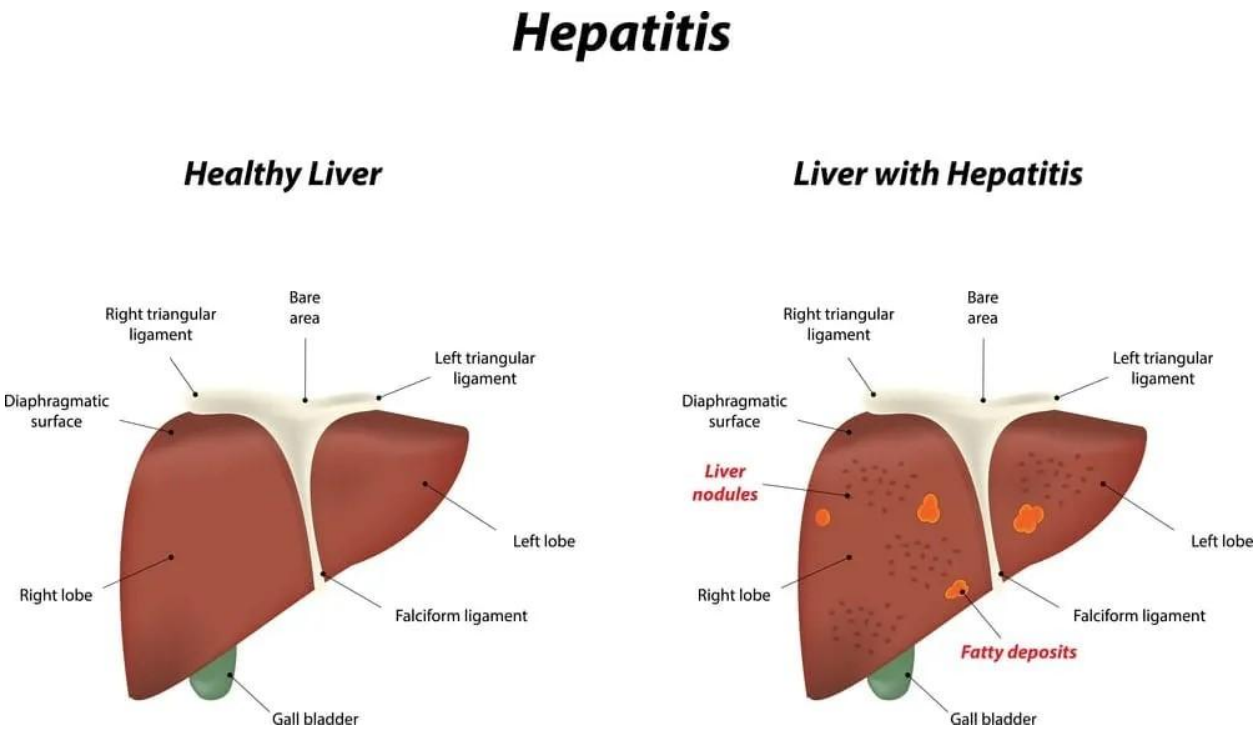

Figure 2-1 Liver Disease and Hepatitis

2.1.2 Mode of Treatment

The Treatment strategies for hepatitis depend on several factors including the type of hepatitis, the severity of the disease, and the presence of any complications. The primary goals of treatment are to reduce liver inflammation, prevent liver damage, and eradicate the virus if possible. Treatment modalities may include antiviral medications, immunosuppressants (in the case of autoimmune hepatitis), lifestyle modifications, and in severe cases, liver transplantation.

Antiviral therapy is a cornerstone of treatment for hepatitis B and C infections. These medications work by inhibiting viral replication, thereby reducing viral load and preventing liver damage. Immunization against hepatitis A and B is also an essential preventive measure, particularly in high-risk populations.

In addition to pharmacological interventions, lifestyle modifications such as avoiding alcohol, maintaining a healthy diet, and regular exercise play a crucial role in managing hepatitis and preventing disease progression. Patients with chronic hepatitis may require long-term monitoring and follow-up to assess treatment response and detect any complications early.

2.2 Artificial Intellegence

Artificial intelligence (AI) is a new discipline that aims to simulate, extend, and expand human intelligence and integrates theory, method, and application research and development (Angermueller et al., 2016). According to the different algorithms of AI, AI is usually divided into traditional AI (e.g., traditional machine learning) and deep learning (based on neural network structure) in the medical field. Compared with traditional AI, deep learning can directly apply the image to the learning process without manual feature extraction, while traditional machine learning needs manual recognition and extraction of different features. Therefore, deep learning has the advantage of no manual extraction of various features, which makes its learning process faster, intelligent, and accurate. In addition, deep learning can iterate and improve from past mistakes, but it needs more big data and more result analysis to fully show its robust and precise efficiency (Zhou et al., 2019).

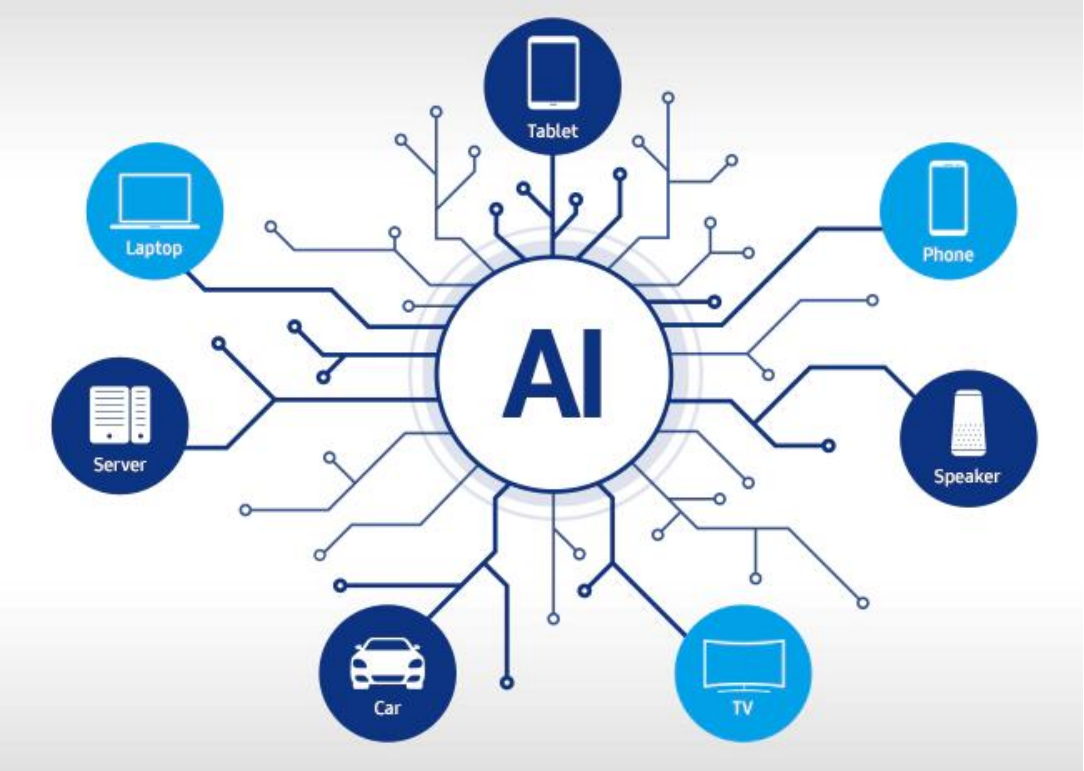

Figure 2-2: Artificial Intelligence (AI).

2.2.1 ANFIS

ANFIS, or Adaptive Neuro-Fuzzy Inference Systems, represents a fusion of two prominent paradigms in artificial intelligence: Artificial Neural Networks (ANN) and Fuzzy Logic (FL). This hybrid approach combines the learning capabilities of neural networks with the reasoning capabilities of fuzzy logic to create a powerful tool for modeling complex systems and making accurate predictions.

Artificial neural networks (ANNs) are biologically inspired computer programs designed to simulate the way in which the human brain processes information. ANNs gather their knowledge by detecting the patterns and relationships in data and learn (or are trained) through experience, not from programming. An ANN is formed from hundreds of single units, artificial neurons or processing elements (PE), connected with coefficients (weights), which constitute the neural structure and are organised in layers. The power of neural computations comes from connecting neurons in a network (Agatonovic et al, 2000).

On the other hand, Fuzzy Logic (FL) provides a framework for dealing with uncertainty and imprecision in data by allowing for the representation of vague or ambiguous concepts. Fuzzy logic extends traditional binary logic by incorporating degrees of truth, enabling more nuanced reasoning in decision-making processes.

2.2.2. Ensemble Method

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. One of the most active areas of research in supervised learning has been to study methods for constructing good ensembles of classifiers. The main discovery is that ensembles are often much more accurate than the individual classifiers that make them up.

A necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse (Hansen et al, 1990). An accurate classifier is one that has an error rate of better than random guessing on new x values. Two classifiers are diverse if they make different errors on new data points. To see why accuracy and diversity are good, imagine that we have an ensemble of three classifiers: {h1, h2, h3} and consider a new case x. If the three classifiers are identical (i.e., not diverse), then when h1(x) is wrong, h2(x) and h3(x) will also be wrong. However, if the errors made by the classifiers are uncorrelated, then when h1(x) is wrong, h2(x) and h3(x) may be correct, so that a majority vote will correctly classify x. More precisely, if the error rates of L hypotheses h are all equal to $p < 1/2$ and if the errors are independent, then the probability that the majority vote will be wrong will be the area under the binomial distribution where more than L/2 hypotheses are wrong.

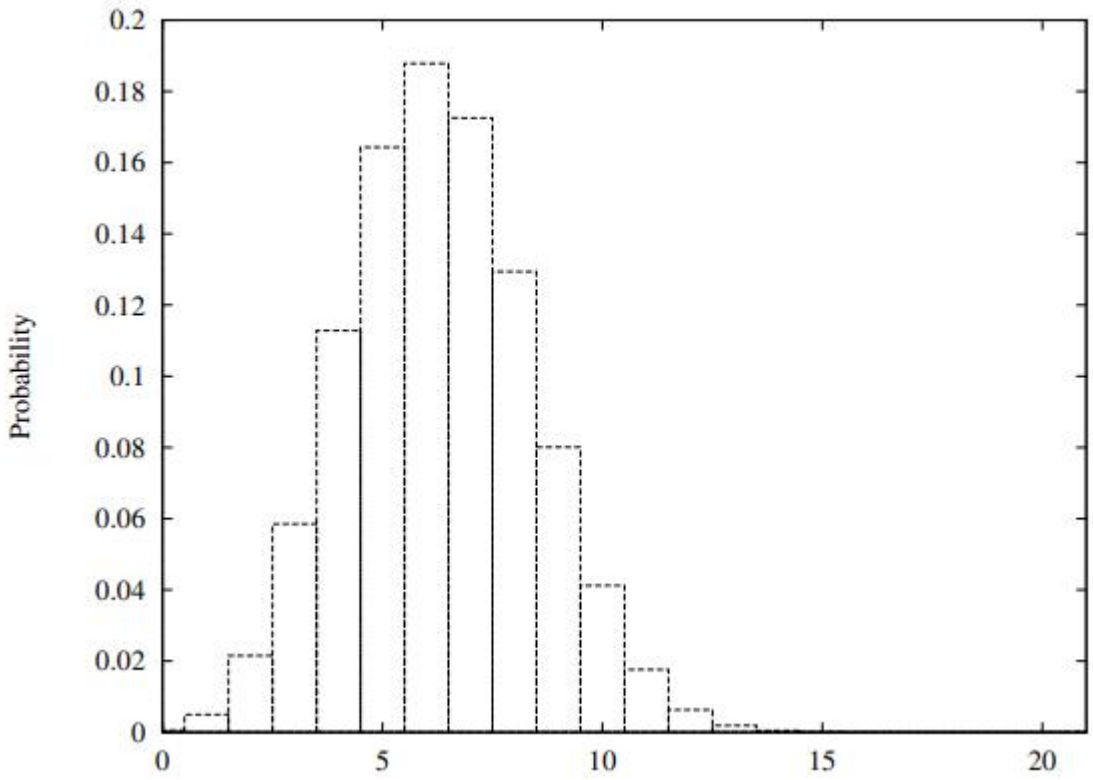

Figure 2- 3 Number of classifiers in error

Figure 2-3 shows this for a simulated ensemble of 21 hypotheses, each having an error rate of 0.3. The area under the curve for 11 or more hypotheses being simultaneously wrong is 0.026, which is much less than the error rate of the individual hypotheses.

2.2.3. Different Methods

One In addition to ANFIS and ensemble methods, a variety of other AI techniques can be applied to hepatitis diagnosis. These include traditional machine learning algorithms such as decision trees, support vector machines, and k-nearest neighbors, as well as more advanced deep learning architectures like convolutional neural networks and recurrent neural networks. Each of these methods has its strengths and limitations, and the choice of technique depends on factors such as the nature of the data, the size of the dataset, and the specific requirements of the application. By exploring a diverse range of AI methods, researchers can identify the most suitable approach for addressing the challenges associated with hepatitis diagnosis and prognosis.

2.3 Feature Selection Method

A feature is an individual measurable property of the process being observed. Using a set of features, any machine learning algorithm can perform classification. In the past years in the applications of machine learning or pattern recognition, the domain of features have expanded from tens to hundreds of variables or features used in those applications. Several techniques are developed to address the problem of reducing irrelevant and redundant variables which are a burden on challenging tasks (Chandrashekar et al, 2014). Feature Selection (variable elimination) helps in understanding data, reducing computation requirement, reducing the effect of curse of dimensionality and improving the predictor performance. In this paper we look at some of the methods found in literature which use particular measurements to find a subset of variables (features) which improves the overall prediction performance.

2.3.1 Filter methods

Filter methods use variable ranking techniques as the principle criteria for variable selection by ordering. Ranking methods are used due to their simplicity and good success is reported for practical applications. A suitable ranking criterion is used to score the variables and a threshold is used to remove variables below the threshold. Ranking methods are filter methods since they are applied before classification to filter out the less relevant variables (Chandrashekar et al, 2014).

2.3.1.1 Merits/Demerits

Merits:

1. **Computational Efficiency:** Filtering methods are computationally efficient as they do not require training the model. They evaluate features independently of each other, making them suitable for large datasets with many features.
2. **Independence from Algorithm:** Since filtering methods do not rely on the performance of a specific algorithm, they are versatile and can be applied across various machine learning algorithms and tasks.

3. **Simplicity:** These methods are straightforward to implement and interpret, as they evaluate features based on predefined criteria without considering interactions between features.

Demerits:

1. **Limited Feature Interaction:** Filtering methods may overlook interactions between features, as they evaluate features independently. This may lead to suboptimal feature subsets, especially when interactions play a significant role in the dataset.
2. **Potential Information Loss:** Filtering methods may discard potentially useful features if they do not meet the predefined criteria. This can result in information loss and reduced model performance.
3. **Suboptimal Performance:** The selected feature subset may not always lead to the best performance for a given machine learning task, as filtering methods do not consider the specific requirements of the model or the dataset.

2.3.2 Wrapper methods

Wrapper methods use the predictor as a black box and the predictor performance as the objective function to evaluate the variable subset. Since evaluating $2N$ subsets becomes a NP-hard problem, suboptimal subsets are found by employing search algorithms which find a subset heuristically. A number of search algorithms can be used to find a subset of variables which maximizes the objective function which is the classification performance (Chandrashekar et al, 2014).

2.3.2.1 Merits/Demerits

Merits:

1. **Optimization for Specific Algorithm:** Wrapper methods select features based on the performance of a specific machine learning algorithm. This optimization can lead to a feature subset that maximizes the performance of the chosen algorithm.

2. **Interactions Consideration:** These methods can capture interactions between features, as they evaluate subsets of features in conjunction with the predictive performance of the model.

3. **Potential for High Accuracy:** Since wrapper methods assess feature subsets based on the performance of the model, they have the potential to achieve high accuracy, especially when the feature space is properly explored.

Demerits:

1. **Computational Cost:** Wrapper methods can be computationally expensive as they require training and evaluating the model for each possible feature subset. This makes them impractical for large datasets or when the feature space is extensive.

2. **Overfitting Risk:** There is a risk of overfitting, particularly when the dataset is small or when the model complexity is high. The selected feature subset may perform well on the training data but poorly on unseen data, leading to over-optimistic results.

3. **Algorithm Sensitivity:** The performance of wrapper methods heavily depends on the choice of the machine learning algorithm used for evaluation. The method may not generalize well to other algorithms or different datasets, limiting its applicability.

2.3.3 Embedded methods

Embedded methods [1], [9], [10] want to reduce the computation time taken up for reclassifying different subsets which is done in wrapper methods. The main approach is to incorporate the feature selection as part of the training process. In Section 2 we mentioned that MI is an important concept but the ranking using MI yielded poor results since the MI between the feature and the class output only was considered (Chandrashekar et al, 2014).

2.3.3.1 Merits/Demerits

Merits:

1. **Integration with Model Training:** Embedded methods integrate feature selection directly into the model training process, optimizing feature selection and model learning simultaneously.

2. **Efficient Use of Resources:** These methods are computationally efficient as they do not require separate feature selection steps. Feature selection is performed as part of the model training process, reducing computational overhead.

3. **Adaptability:** Embedded methods automatically adapt to the characteristics of the dataset and the complexity of the learning task, making them suitable for a wide range of applications.

Demerits:

1. **Algorithm Dependency:** Embedded methods are specific to certain machine learning algorithms that support feature selection during training. This limits their applicability to algorithms that do not natively support embedded feature selection.

2. **Risk of Overfitting:** The selected features may overfit to the training data, particularly if the model complexity is high or the dataset is small. This can lead to reduced generalization performance on unseen data.

3. **Hyperparameter Tuning:** Fine-tuning hyperparameters may be required to optimize the performance of embedded feature selection, adding complexity to the model training process.

2.4 ML Algorithm

According to Arthur Samuel Machine learning is defined as the field of study that gives computers the ability to learn without being explicitly programmed. Arthur Samuel was famous for his checkers playing program. Machine learning (ML) is used to teach machines how to handle the data more efficiently. Sometimes after viewing the data, we cannot interpret the extract information from the data. In that case, we apply machine learning. With the abundance of datasets available, the demand for machine learning is in rise. Many industries apply machine learning to extract relevant data. The purpose of machine learning is to learn from the data. Many studies have been done on how to make machines learn by themselves without being explicitly programmed. Many mathematicians and programmers apply several approaches to find the solution of this problem which are having huge data sets. Machine Learning relies on different algorithms to solve data problems. Data scientists like to point out that there's no single one-size-

fits-all type of algorithm that is best to solve a problem. The kind of algorithm employed depends on the kind of problem you wish to solve, the number of variables, the kind of model that would suit it best and so on. Here's a quick look at some of the commonly used algorithms in machine learning (ML) (Batta, 2019).

2.4.1 Support Vector Machine

Another most widely used state-of-the-art machine learning technique is Support Vector Machine (SVM). In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. It basically, draw margins between the classes. The margins are drawn in such a fashion that the distance between the margin and the classes is maximum and hence, minimizing the classification error.

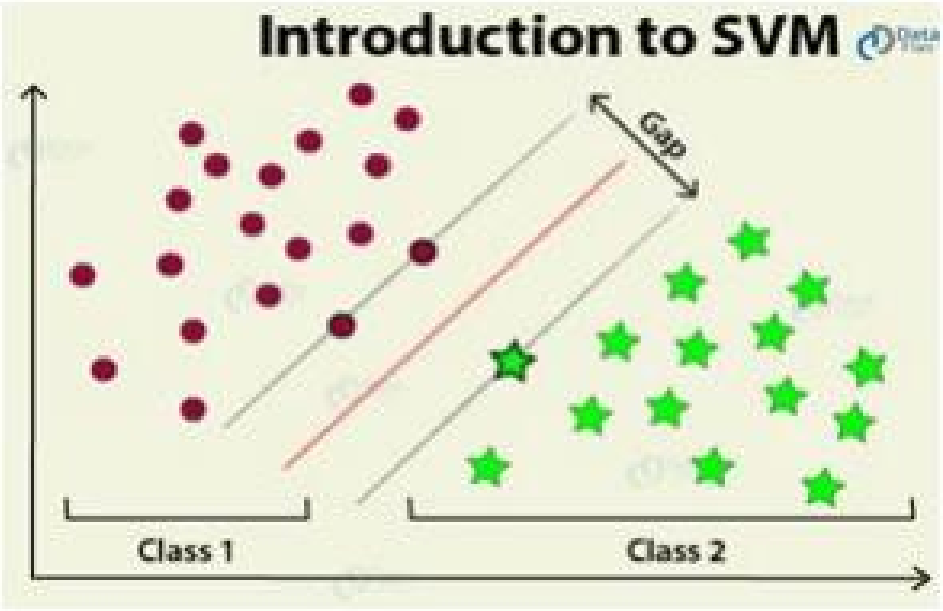

Figure 2-4 Support Vector Machine

2.4.2 Supervised Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. The supervised machine learning algorithms are those

algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. All algorithms learn some kind of patterns from the training dataset and apply them to the test dataset for prediction or classification (Batta, 2019).

Table 2- 1: Review of related Literature

S/N	AUTHOR	OBJECTIVE	METHOD	DATA SOURCE	CONTRIBUTION TO KNOWLEDGE	LIMITATION
1	Rahmon et al, 2018	Diagnosis of hepatitis using ANFIS	FL, ANN	Carnegie Mellon university database	After the system was evaluated, the performance 0metric 0gave accuracy of 90.2%.	Limited data for research
2	Omursal et al, 2015	An application of multilayer neural network on hepatitis using approximation of sigmoid activation function	LM algorithm, MLNN	UCI	The method of classifying hepatitis disease produced an accuracy of 91.9% to 93.8% via 10 fold cross validation.	Difficulty in calculating the exponential expression of the sigmoid activation function.
3	Aqued and alaakhalaf, 2019	Liver hepatitis diagnosing based on fuzzy inference	FIS	UCI	Diagnosis accuracy reaches 96%, with the	Difficulty in removing the noise before applying the

		system			corresponding approximated time ranging between 0.10 ↄ 0.15 seconds.	visual feature.
4	Dalwinder et al, 2021	A multi layered fuzzy inference system for diagnosis of hepatitis B	MLFIS	School of CSE	Various parameters used to calculate the performance of the system are also determined. The classification accuracy of this system is 94%	Lack of sufficient data for research
5	Khosro et al, 2012	An intelligent diagnostic system for detection of hepatitis using multi layer perception and colonial competitive	CCA, MLP	UCI	The 0.0163, 0.0454, 98.38% and 95.46% values presents the mean square of error for the data used in training and testing system	Inability to manage very large neutrons.

		algorithm.			respectively as well as the ultimate accuracy of the program in data training and testing.	
6	Waheed et al, 2019	Intelligent hepatitis diagnosis using ANFIS and information gain method	IG, ANFIS	UCI	The performance of the proposed approach was evaluated using statistical methods, and the highest results for the classification accuracy, specificity, and sensitivity analysis of the proposed system reached were 95.24%, 91.7%, and 96.17%,	Limited data to research on.

					respectively	
7	Mina et al,2019	Hybrid adaptive Neuro fuzzy inference system for diagnosing liver disorders	ANFIS- PCO	UCI	With this recommended strategy, the accuracy of classification is increased by 10% compared with the ANFIS based on the dataset can be accomplished.	
	Proposal lacks generalization					
8	Abbas et al,2022	An adaptive Neuro fuzzy inference system and principal component analysis: A hybridized method for viral hepatitis diagnosis system	PCA- ANFIS	UCI	The hybrid PCA-ANFIS method is achieved remarkable higher Accuracy (97.41%) compared with the state-of-the-art, and it can be an	

					adopted tool for diagnosing Hepatitis.	
	Proposal lacks generalization					
9	Adetokunbo et al,2020	A survey on artificial intelligence based techniques for diagnosis of hepatitis variants	ANN, FL	Secondary data from already published papers from open access journals over the internet	Results showed that Hepatitis B (30%) and C (3%) were the only types of hepatitis the AI-based techniques were used to diagnose and properly classified out of the five major types, while (67%) of the paper reviewed diagnosed hepatitis disease based on	Other type of hepatitis were left out by the articles and they focused on only hepatitis B.

					the different AI based approach but were not classified into any of the five major types.	
10	Sara et al,2020	Modelling for predicting the severity of hepatitis based on artificial neural networks	BPNN RBFN KMN	UCI	Diagnosis accuracy was 99.33% for 2-class and 88% for 5-class, which considered excellent accuracy.	
	No feature selection					
11	Set dal and fetzullah, 2011	A study on hepatitis diseases diagnosis using multi layer neural network with lavenberg Marquardt training algorithm	MLNN	UCI	We obtained a classification accuracy of 91.87% via tenfold cross validation.	Suffers slow convergence rate and often yields suboptimal solutions.

12	Mehrbakhsh et al, 2019	A predictive method for hepatitis disease diagnosis using ensembls of Neuro fuzzy techniques	NIPALs, SOM CART ANFIS	UCI	The accuracy of our method on this dataset measured by ROC was 93.06%	Proposal lacks generalization
13	Sinan et al,2022	Fuzzy logic and correlation based hybrid on hepatitis disease dataset	FL	UCI	To estimate hepatitis disease based on rough set attributes selection algorithm helping the health worker in making right decision.	It has problems finding suitable membership values .
14	Alaa et al, 2019	Diagnosis of hepatitis disease with logistic regression and artificial neural network	LR, MLPNN, ANN	UCI	Diagnosis of the hepatitis dataset with 97.3% AUCROC for the training subset and 88.6% for the test one.	Lacked enough data to conduct proper research

LEGEND

1. FL: Fuzzy logic
2. ANN: Adaptive neural network
3. LM: Levenberg Morquardt
4. MLNN: Multi-layer neural network
5. FIS: Fuzzy inference system
6. MLFIS: Multi-layer fuzzy inference system
7. CCA: Colonial competitive algorithm
8. MLP: Multi-layer perception
9. ANFIS: Adaptive Neuro fuzzy inference system
10. ANFIS-PSO: Adaptive neuro fuzzy inference system- particle swarm optimization.
11. PCA-ANFIS: principal component analysis- Adaptive neuro fuzzy inference system
12. BPNN: Back propagation neural network.
13. RBFN: Radial basis function network.
14. KNN: K-nearest neighbor
15. NIPALS: Nonlinear iterative partial least squares.
16. SOM: Self-organizing map
17. LR: Logistic regression
18. MLPNN: Multilayer perceptual neural network

CHAPTER 3

RESEARCH METHODOLOGY

In this chapter, we detail the methodology employed to develop and evaluate the predictive model for hepatitis diagnosis using ensemble feature selection techniques and Adaptive Neuro-Fuzzy Inference Systems (ANFIS). The methodology follows the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework, encompassing several stages: data collection, data preprocessing, ensemble feature selection, model development, and model evaluation. Each stage is crucial for the successful implementation and validation of the proposed predictive model.

3.1 Data Collection

The initial step in our study is to gather the hepatitis dataset from a reliable and relevant source. Data collection plays a pivotal role in ensuring the quality and representativeness of the dataset, which is crucial for developing an accurate predictive model for hepatitis diagnosis.

3.1.1 Source Selection:

For this study, we selected the hepatitis dataset available on Kaggle, a popular platform for sharing datasets and conducting data-driven research. The dataset was chosen for its comprehensive information on patients diagnosed with hepatitis, including demographic details, clinical attributes, and laboratory test results.

3.1.2 Dataset Description:

The hepatitis dataset comprises structured data collected from patients diagnosed with hepatitis. It includes various features such as age, sex, symptoms, laboratory test results (e.g., Bilirubin levels, Albumin levels), and the target variable indicating the hepatitis category (e.g., Type A, Type B).

3.1.3 Data Relevance:

The dataset's relevance to our research objectives lies in its focus on hepatitis diagnosis, making it suitable for training and evaluating predictive models for hepatitis classification. By

encompassing diverse patient demographics and clinical attributes, the dataset offers valuable insights into the factors influencing hepatitis diagnosis and prognosis.

3.1.4 Data Integrity:

Ensuring data integrity is paramount to the success of our study. We meticulously reviewed the dataset to identify any inconsistencies, missing values, or data quality issues. Steps were taken to address these issues through data preprocessing techniques, thereby enhancing the dataset's reliability and usability for model development.

3.1.5 Data Privacy and Ethics:

Respecting patient privacy and adhering to ethical standards are fundamental principles in healthcare-related research. We ensured that the hepatitis dataset used in our study complies with relevant data protection regulations and guidelines. Patient identifiers were removed or anonymized to preserve confidentiality and prevent unauthorized access to sensitive information.

3.2 Data Preprocessing

Data preprocessing is a crucial step in preparing the raw dataset for analysis and model training. It involves various techniques to clean, transform, and enhance the data, ensuring its quality and compatibility with machine learning algorithms. In this section, we outline the steps taken to preprocess the hepatitis dataset before model development.

3.2.1 Handling Missing Values:

The first task in data preprocessing is to address missing values, which can adversely affect model performance. We identified missing values in the dataset, represented as '?' and replaced them with appropriate values or employed imputation techniques such as mean or median imputation for numerical features and mode imputation for categorical features.

3.2.2 Encoding Categorical Variables:

Many machine learning algorithms require numerical input, necessitating the encoding of categorical variables. We utilized Ordinal Encoding to convert categorical variables, such as 'Sex' and 'Category' (target variable), into numerical representations. This transformation enables the algorithms to process these variables effectively during model training.

3.2.3 Feature Scaling:

To ensure uniformity in feature magnitudes and prevent certain features from dominating others, we performed feature scaling. Numerical features were standardized to have a mean of 0 and a standard deviation of 1, thereby bringing them to a similar scale. This step facilitates convergence during model training and enhances algorithm performance.

3.2.4 Feature Engineering:

Feature engineering involves creating new features or transforming existing ones to improve model accuracy. We explored feature engineering techniques such as polynomial features, interaction terms, and domain-specific transformations to capture complex relationships and enhance the dataset's predictive power.

3.2.5 Data Splitting:

Before model development, we split the preprocessed dataset into training and testing sets. The training set is used to train the model, while the testing set evaluates its performance on unseen data. We employed stratified sampling to ensure balanced representation of target classes in both training and testing sets, thereby preventing class imbalance issues.

3.3 Ensemble Feature Selection

Ensemble feature selection is a critical phase aimed at identifying the most informative features that contribute significantly to the prediction task while eliminating irrelevant or redundant features. In this section, we detail the ensemble feature selection process employed to identify relevant features for hepatitis diagnosis.

3.3.1 ReliefF:

ReliefF is a feature selection algorithm that evaluates the relevance of each feature by considering its ability to distinguish between instances of the same and different classes. By iteratively sampling instances and updating feature weights, ReliefF assigns higher weights to features that have a greater impact on classification accuracy. We implemented ReliefF to rank features based on their importance and selected the top-ranked features for further analysis.

3.3.2 Fisher's Scores:

Fisher's scores, also known as Fisher discriminant ratio, measure the discriminative power of features by calculating the ratio of between-class variance to within-class variance. Features with higher Fisher's scores indicate greater discrimination between classes and are considered more relevant for classification. We computed Fisher's scores for each feature and selected those with the highest scores as potential candidates for feature selection.

Fisher score is one of the most widely used supervised feature selection methods. The algorithm we will use returns the ranks of the variables based on the fisher's score in descending order. We can then select the variables as per the case (Gupta, 2023).

```
from skfeature.function.similarity_based import fisher_score
import matplotlib.pyplot as plt
%matplotlib inline

# Calculating scores
ranks = fisher_score.fisher_score(X, Y)

# Plotting the ranks
feat_importances = pd.Series(ranks, dataframe.columns[0:len(dataframe.columns)-1])
feat_importances.plot(kind='barh', color = 'teal')
plt.show()
```

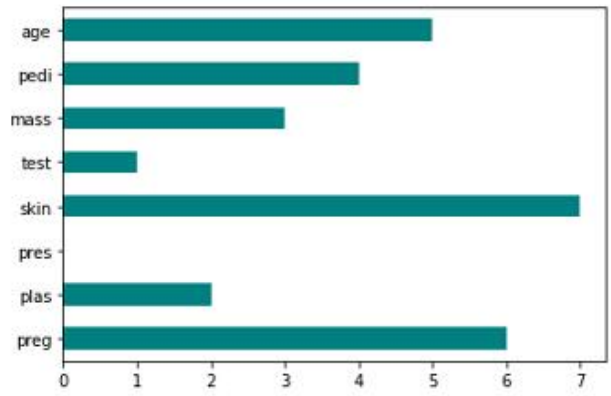

Figure 3-1 Fisher's Score

3.3.3 Correlation:

Correlation analysis assesses the linear relationship between features and the target variable. Features with high correlation coefficients with the target variable are likely to be more predictive and informative for classification tasks. We computed Pearson correlation coefficients

between each feature and the target variable and selected features with the highest absolute correlation values.

3.3.4 Ensemble Selection:

To obtain a robust set of features, we employed ensemble selection techniques that combine the results of multiple feature selection methods. By aggregating the rankings or scores obtained from ReliefF, Fisher's scores, and correlation analysis, we identified features that consistently ranked high across different methods. This ensemble approach ensures the selection of features with diverse characteristics and improves the overall robustness of the feature selection process.

3.4 Model Development

In this section, we outline the process of developing the predictive model for hepatitis diagnosis using an Adaptive Neuro-Fuzzy Inference System (ANFIS). ANFIS combines the strengths of neural networks and fuzzy logic to create a hybrid model capable of handling complex and uncertain data.

3.4.1 Model Architecture:

The ANFIS model consists of multiple layers, each serving a specific purpose in the learning process. The architecture includes:

1. Input Layer: Receives input features representing patient attributes such as age, sex, and laboratory test results.
2. Fuzzification Layer: Converts crisp input values into fuzzy sets using linguistic terms such as "young," "middle-aged," or "elderly" for age.
3. Rule Base: Defines fuzzy rules that capture relationships between input features and output categories (hepatitis diagnosis).
4. Inference Engine: Computes the degree of membership for each input in fuzzy sets and evaluates the firing strengths of rules.
5. Defuzzification Layer: Aggregates the outputs of rules to generate crisp output values representing the predicted diagnosis.

3.4.2 Model Training:

The ANFIS model is trained using a hybrid learning algorithm that combines gradient descent and least squares methods. During training, the parameters of fuzzy membership functions and rule antecedents are optimized to minimize the error between predicted and actual diagnoses. The training process involves iteratively adjusting the model parameters using backpropagation and gradient descent techniques until convergence is achieved.

3.4.3 Hyperparameter Tuning:

To optimize the performance of the ANFIS model, hyperparameters such as the number of membership functions, epochs, and learning rate are fine-tuned through cross-validation. Hyperparameter tuning aims to find the optimal configuration that maximizes model accuracy and generalization while avoiding overfitting.

Hyperparameter tuning methods

1. **Random Search**

   In the random search method, we create a grid of possible values for hyperparameters. Each iteration tries a random combination of hyperparameters from this grid, records the performance, and lastly returns the combination of hyperparameters that provided the best performance.

2. **Grid Search**
3. In the grid search method, we create a grid of possible values for hyperparameters. Each iteration tries a combination of hyperparameters in a specific order. It fits the model on each and every combination of hyperparameters possible and records the model performance. Finally, it returns the best model with the best hyperparameters (Es, 2023).

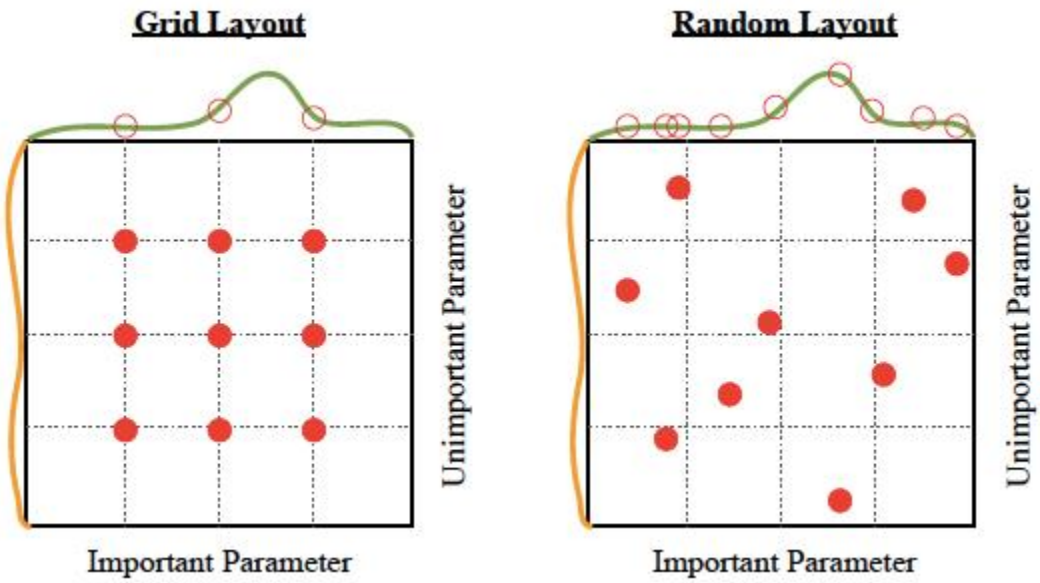

Figure 3-2 Grid Search

3.3.4 Model Evaluation:

Once trained, the ANFIS model is evaluated using a held-out test dataset to assess its predictive performance. Performance metrics such as accuracy, precision, recall, and F1-score are computed to quantify the model's ability to correctly classify hepatitis cases. Additionally, confusion matrices and ROC curves may be generated to visualize the model's classification performance and assess its sensitivity and specificity.

3.5 Model Evaluation:

After developing the ANFIS model, it is crucial to evaluate its performance to assess its effectiveness in predicting hepatitis cases accurately. Model evaluation involves assessing various performance metrics and analyzing the model's behavior across different datasets. The following steps outline the process of evaluating the ANFIS model:

3.5.1 Performance Metrics:

Accuracy: Measures the overall correctness of the model's predictions.

Precision: Indicates the proportion of true positive predictions among all positive predictions.

Recall (Sensitivity): Measures the proportion of actual positives that were correctly identified by the model.

F1-Score: Harmonic mean of precision and recall, providing a balance between the two metrics.

Confusion Matrix: Tabulates true positive, true negative, false positive, and false negative predictions to assess classification performance.

3.5.2 Cross-Validation:

Utilize techniques such as k-fold cross-validation to validate the model's performance across multiple subsets of the dataset. This helps ensure that the model's performance is consistent and not heavily influenced by the particular training-test split.

3.5.3 Test Dataset Evaluation:

Evaluate the trained ANFIS model on a held-out test dataset that was not used during training. This provides an unbiased estimate of the model's generalization performance on unseen data.

3.5.4 Visualizations:

Generate visualizations such as ROC curves to visualize the trade-off between true positive rate and false positive rate at different classification thresholds.

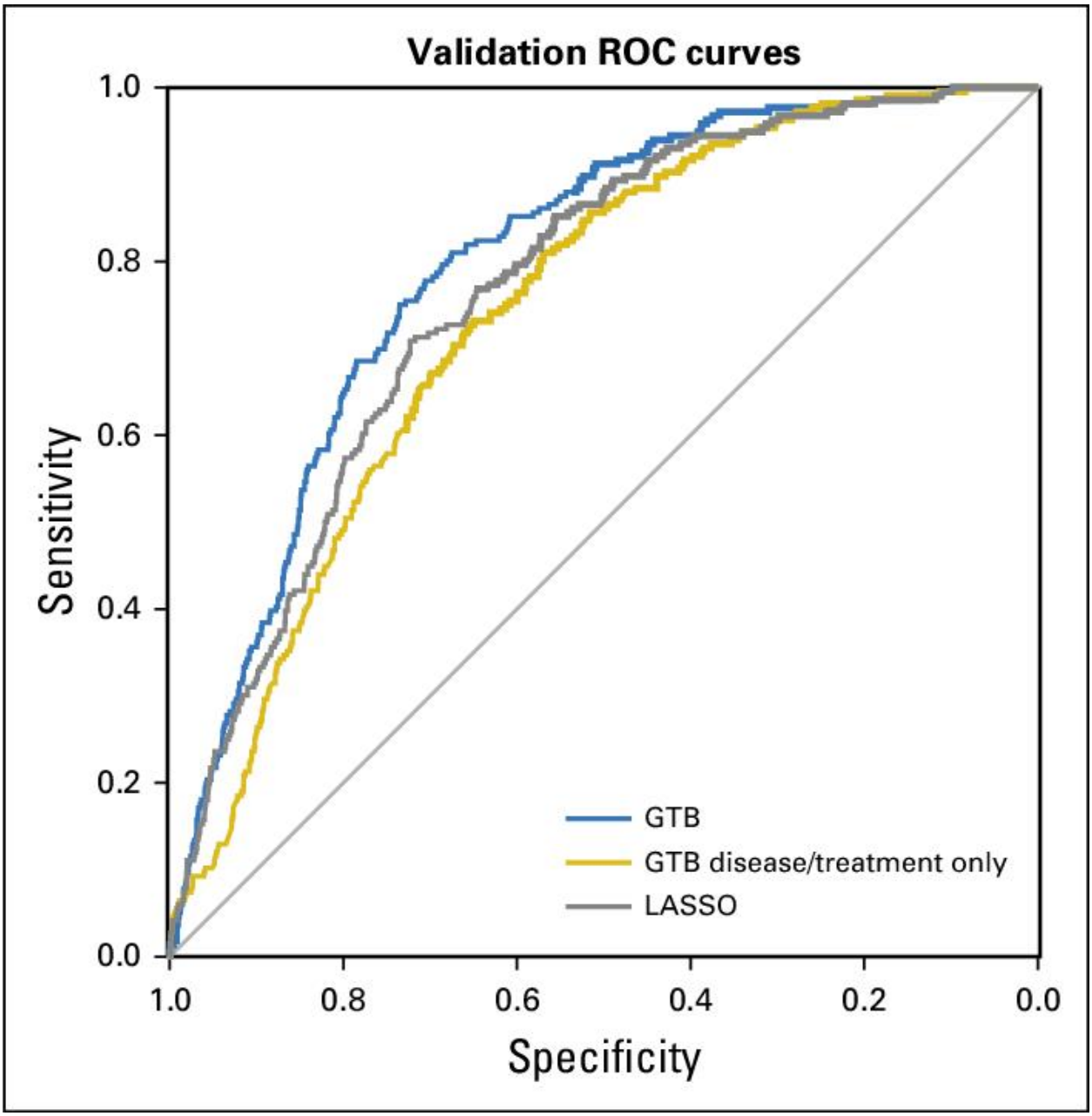

Figure 3-3 ROC curve

Plot learning curves to analyze the model's performance as a function of training data size and epochs. This helps identify potential overfitting or underfitting issues.

3.5.5 Interpretability:

Assess the interpretability of the ANFIS model by analyzing the learned fuzzy rules and membership functions. Understandable rules and transparent decision-making processes enhance the model's trustworthiness and usability in clinical settings.

3.5.6 Comparative Analysis:

Compare the performance of the ANFIS model with other machine learning algorithms or traditional diagnostic methods commonly used in hepatitis diagnosis. This provides insights into the relative strengths and weaknesses of different approaches.

3.5.7 Sensitivity Analysis:

Conduct sensitivity analysis to evaluate the robustness of the ANFIS model to variations in input data or model parameters. This helps identify potential sources of uncertainty and assess the model's stability.

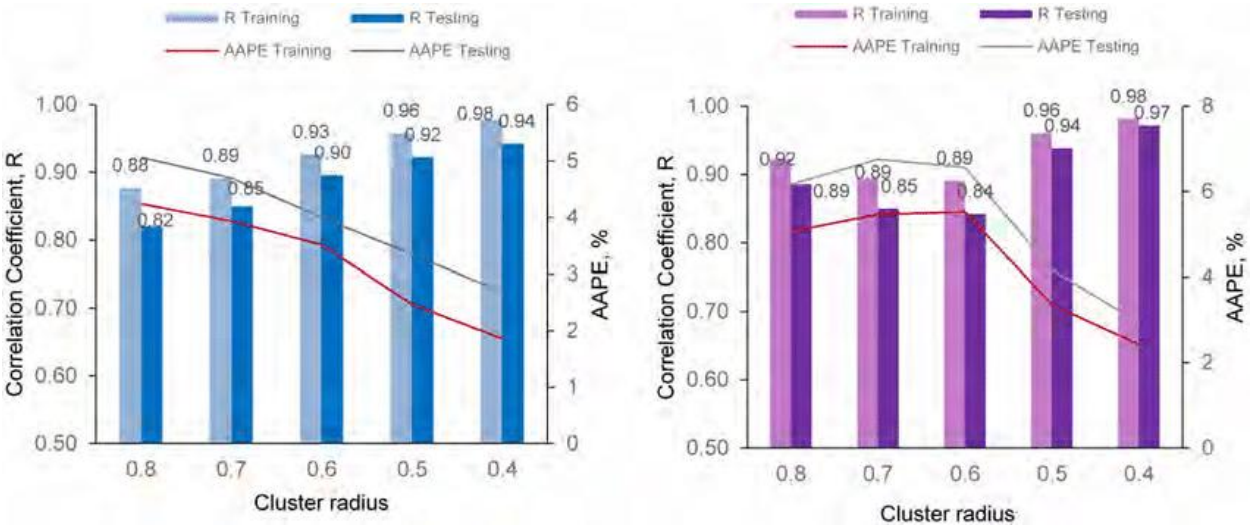

Figure 3-4 Sensitivity analysis of ANFIS model for compressional (on left) and shear (on right) wave slowness.

CHAPTER 4

IMPLEMENTATION AND RESULTS

In this chapter, we delve into the practical aspects of implementing the research methodology outlined in Chapter 1. We document the steps taken to preprocess the data, perform ensemble feature selection, develop the ANFIS model, evaluate its performance, and present the findings obtained from the experimentation process.

4.1 System Requirements

Before proceeding with the implementation details, it's crucial to outline the system requirements necessary to execute the project effectively.

4.1.1 Hardware Requirements

The project's hardware requirements include:

- A processor: An Intel i5 processor or equivalent, or higher, to ensure efficient computation.
- RAM: A minimum of 12GB of RAM is recommended to handle the computational demands effectively.
- Storage: At least 12GB of storage space is required to store project files, datasets, and model checkpoints.
- Processor Speed: A CPU clock speed of 2.5GHz or higher is recommended for faster processing of data.
- Graphics Processing Unit (GPU): Having a compatible GPU is essential for enhanced performance during complex computations and deep learning model training.
- Tensor Processing Unit (TPU): Utilizing a TPU can accelerate the training of machine learning models for faster computation.
- Internet Connection: A stable and reliable internet connection is necessary for accessing online services and libraries required for the project.

4.1.2 Software Requirements

The necessary software requirements for running the project include:

- Operating System: The project is compatible with Windows 7, 8, or 10, and Linux distributions, with Linux being recommended for compatibility and ease of use.
- Integrated Development Environment (IDE): Google Colaboratory (Google Colab) is chosen as the IDE due to its cloud-based Python programming and machine learning tools, providing a collaborative and interactive environment for development.

4.2 Tools and Libraries

In the development of the project, several tools and software libraries are utilized to facilitate the creation, training, and evaluation of machine learning models.

4.2.1 Programming Languages Used

The primary programming language employed in the project is Python, along with the following libraries:

- NumPy: Providing support for large, multi-dimensional arrays and matrices, essential for data manipulation and scientific computing.
- Pandas: Offering data structures and functions for working with structured data, instrumental in data preprocessing and analysis.
- Seaborn and Matplotlib: Data visualization libraries used for creating informative and visually appealing statistical graphics and plots.
- Scikit-Learn: A powerful machine learning library providing a wide range of tools for classification, regression, clustering, and more, simplifying the implementation and evaluation of machine learning models.

4.2.2 Integrated Development Environment (IDE)

Google Colaboratory (Google Colab) serves as the integrated development environment (IDE) for the project, offering a cloud-based platform with free access to GPU and TPU resources, making it suitable for machine learning tasks.

4.3 Data Preprocessing

Data preprocessing is a crucial step in any machine learning project, as it involves cleaning and transforming raw data into a format suitable for model training. In this section, we detail the steps taken to preprocess the hepatitis dataset before feature selection and model development.

4.3.1 Handling Missing Values

The first step in data preprocessing is handling missing values, as they can adversely affect model performance. In the hepatitis dataset, missing values are denoted by '?'. We replace these missing values with NaN (Not a Number) using the **replace()** function, and then use the **dropna()** function to remove rows with missing values.

```python
# Handle missing values
data.replace('?', np.nan, inplace=True)
data.dropna(inplace=True)
```

Figure 4-1 Handle missing values

4.3.2 Encoding Categorical Variables

Machine learning models require numerical inputs, so categorical variables need to be encoded into numerical format. We use ordinal encoding to convert categorical variables ('Category' and 'Sex') into numerical values.

```python
# Encode categorical variables
ordinal_encoder = OrdinalEncoder()
data[['Category', 'Sex']] = ordinal_encoder.fit_transform(data[['Category', 'Sex']])
```

Figure 4-2 Encode categorical variables

4.3.3 Convert Target Variable to Numerical

The target variable ('Category') is also categorical, so we convert it to numerical format to facilitate model training.

```python
# Convert target variable to numerical
data['Category'] = data['Category'].astype(int)
```

Figure 4-3 Convert target variable to numerical

4.3.4 Data Visualization

Before proceeding further, it's essential to visualize the distribution of the target variable to gain insights into the dataset. We use seaborn's **countplot()** function to plot the distribution.

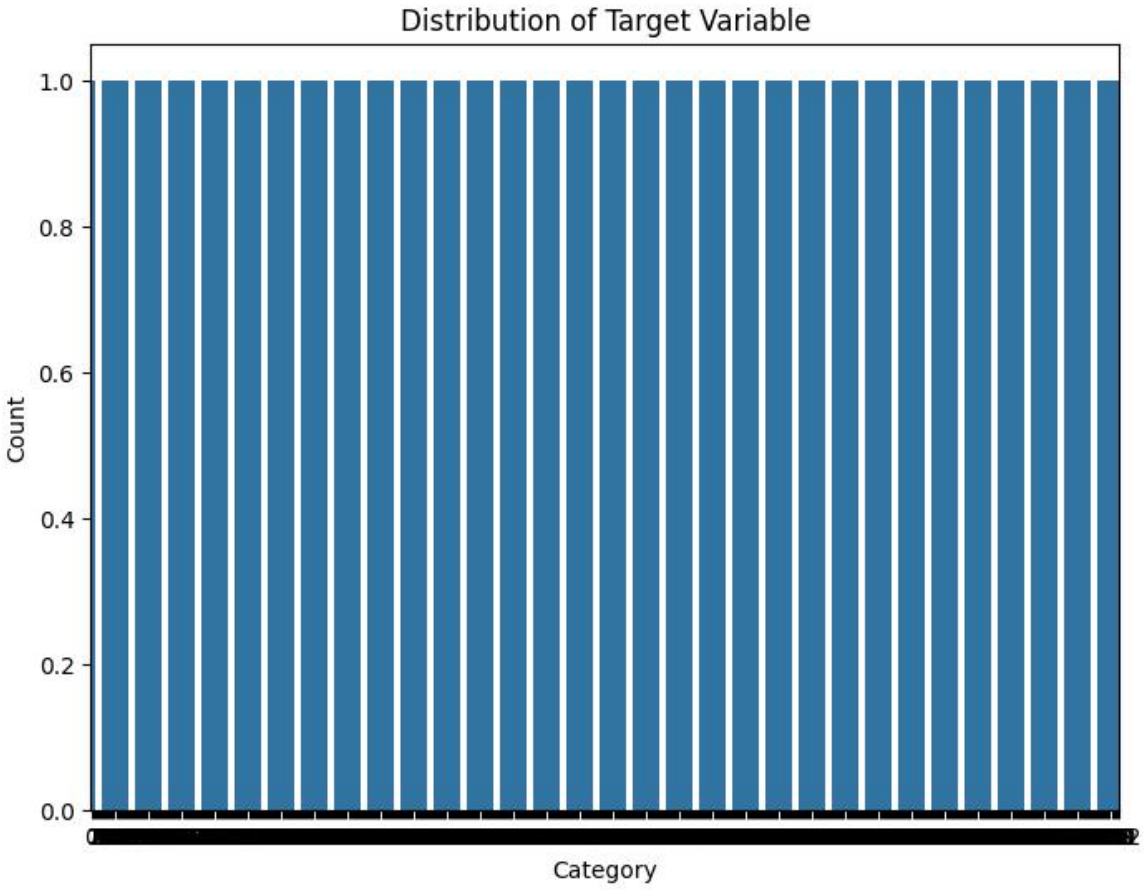

Figure 4-4 Distribution of Target Variable

These preprocessing steps ensure that the dataset is clean, formatted correctly, and ready for feature selection and model development.

4.4 Ensemble Feature Selection

Feature selection plays a critical role in machine learning model development by identifying the most relevant features that contribute to predictive accuracy. In this section, we describe the ensemble feature selection process employed to identify informative features related to hepatitis diagnosis.

4.4.1 ReliefF, Fisher's Scores, and Correlation

To identify the most informative features related to hepatitis diagnosis, we employ a combination of ReliefF, Fisher's scores, and Correlation techniques. These ensemble feature selection

methods help in selecting features that have the highest impact on predicting hepatitis cases accurately.

ReliefF: ReliefF is a feature selection algorithm that evaluates the relevance of features by considering both the distance between instances and their class labels. It assigns scores to features based on how well they discriminate between different classes.

Fisher's Scores: Fisher's score, also known as Fisher Discriminant Analysis, is a statistical technique used for feature selection. It measures the ratio of between-class variance to within-class variance, identifying features that are most discriminative between classes.

Correlation: Correlation analysis measures the strength and direction of the linear relationship between two variables. By calculating the correlation coefficients between features and the target variable, we can identify features that are highly correlated with hepatitis diagnosis.

Implementation

We use the **SelectKBest** method from scikit-learn, along with the ANOVA F-value for ReliefF, to select the top k features based on their importance scores. The selected features are then used for model training and evaluation.

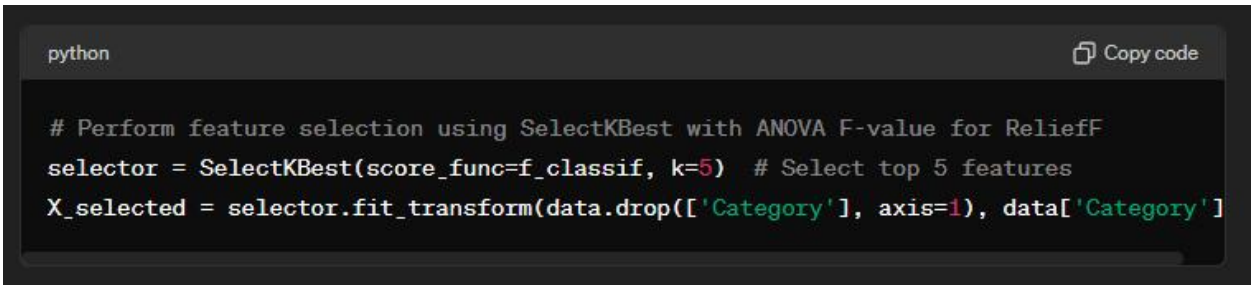

Figure 4-5 Perform feature selection using SelectKBest with ANOVA F-value for ReliefF

Visualization

We visualize the importance scores of the selected features using a bar plot to provide insights into which features contribute the most to hepatitis diagnosis.

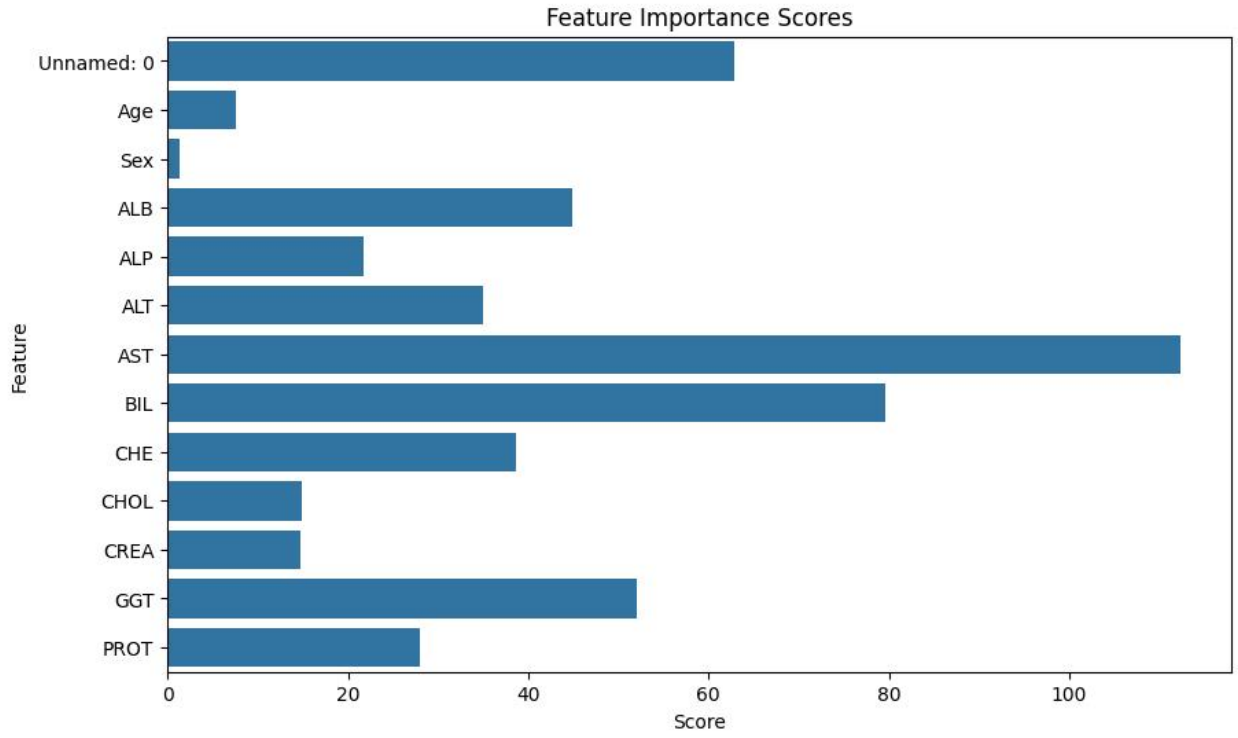

Figure 4-6 Feature Selection Visualization

These ensemble feature selection techniques help in identifying the most relevant features, which are essential for developing an accurate predictive model for hepatitis diagnosis.

4.5 Model Development

Model development is a crucial phase in the project, where we build the predictive model using the selected features to accurately classify hepatitis cases. In this section, we outline the steps involved in developing the ANFIS (Adaptive Neuro-Fuzzy Inference System) model using TensorFlow.

4.5.1 ANFIS Model

The ANFIS model combines the capabilities of neural networks and fuzzy logic to create a powerful predictive system. It leverages fuzzy inference systems to capture linguistic information from data and adaptively tune its parameters using neural network learning algorithms.

Implementation

We initialize and train the ANFIS model using TensorFlow, which provides a robust framework for building and training neural networks.

```python
# Initialize ANFIS model
anfis_model = ANFIS(num_mfs=5, epochs=1000, learning_rate=0.001)

# Train the ANFIS model
anfis_model.fit(X_train, y_train)
```

Figure 4-7 Initialize ANFIS model

Prediction

Once the model is trained, we use it to make predictions on the test dataset.

```python
# Predict using ANFIS model
y_pred = anfis_model.predict(X_test)
y_pred_binary = (y_pred > 0.5).astype(int)
```

Figure 4-8 Predict using ANFIS model

Accuracy Evaluation

We evaluate the accuracy of the ANFIS model by comparing its predictions with the actual labels from the test dataset.

Visualization

To visualize the training progress and assess the model's convergence, we plot the accuracy over epochs.

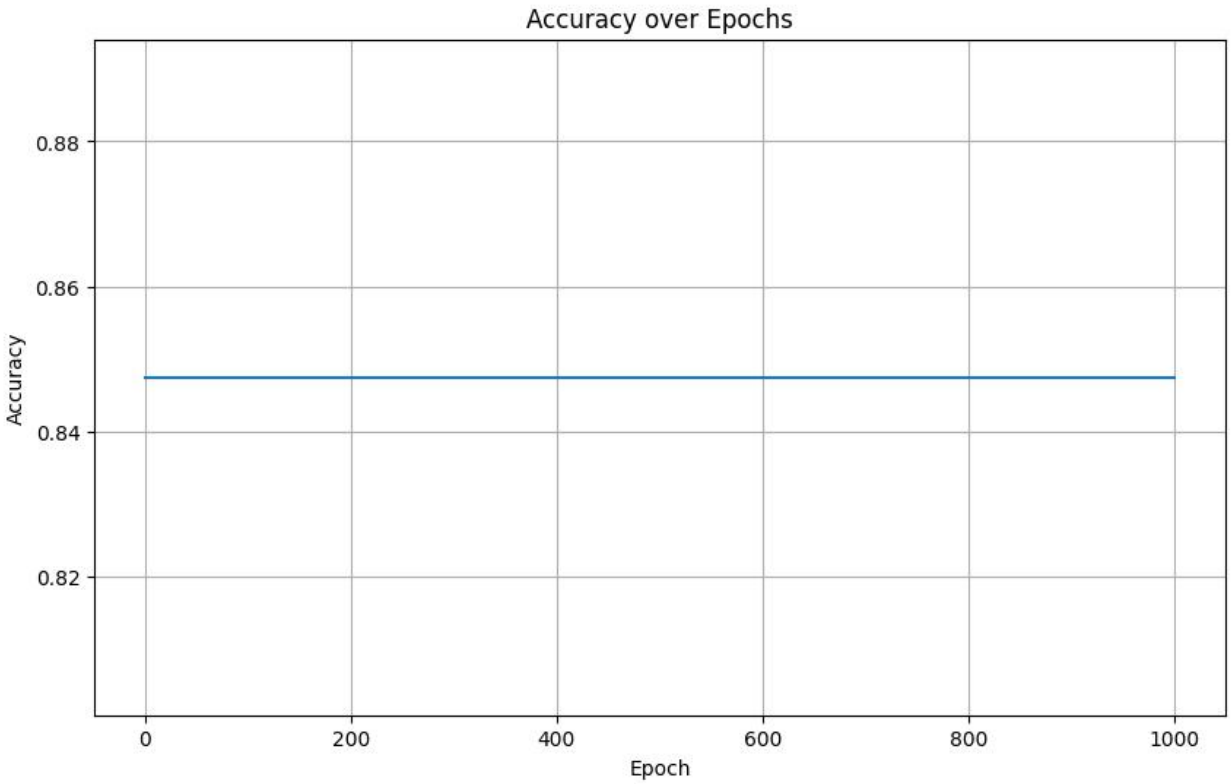

Figure 4-9 Plot accuracy over epochs

The ANFIS model development process ensures that we have a robust and accurate predictive model for hepatitis diagnosis, leveraging both neural network techniques and fuzzy logic to capture complex relationships in the data.

4.6 Model Evaluation

Model evaluation is essential to assess the performance and effectiveness of the developed ANFIS model in predicting hepatitis cases accurately. In this section, we describe the methodology used for evaluating the model's performance and present the results obtained.

4.6.1 Performance Metrics

Various performance metrics are computed to evaluate the ANFIS model's effectiveness in predicting hepatitis cases. These metrics provide insights into different aspects of the model's performance, including accuracy, precision, recall, F1-score, and AUC-ROC.

Implementation

We compute these metrics using appropriate functions available in Python libraries such as Scikit-Learn.

```python
# Compute performance metrics
accuracy = accuracy_score(y_test, y_pred_binary)
precision = precision_score(y_test, y_pred_binary)
recall = recall_score(y_test, y_pred_binary)
f1 = f1_score(y_test, y_pred_binary)
```

Figure 4-10 Compute performance metrics

Visualization

To provide a comprehensive view of the model's performance across multiple metrics, we present the results using visualizations such as bar charts.

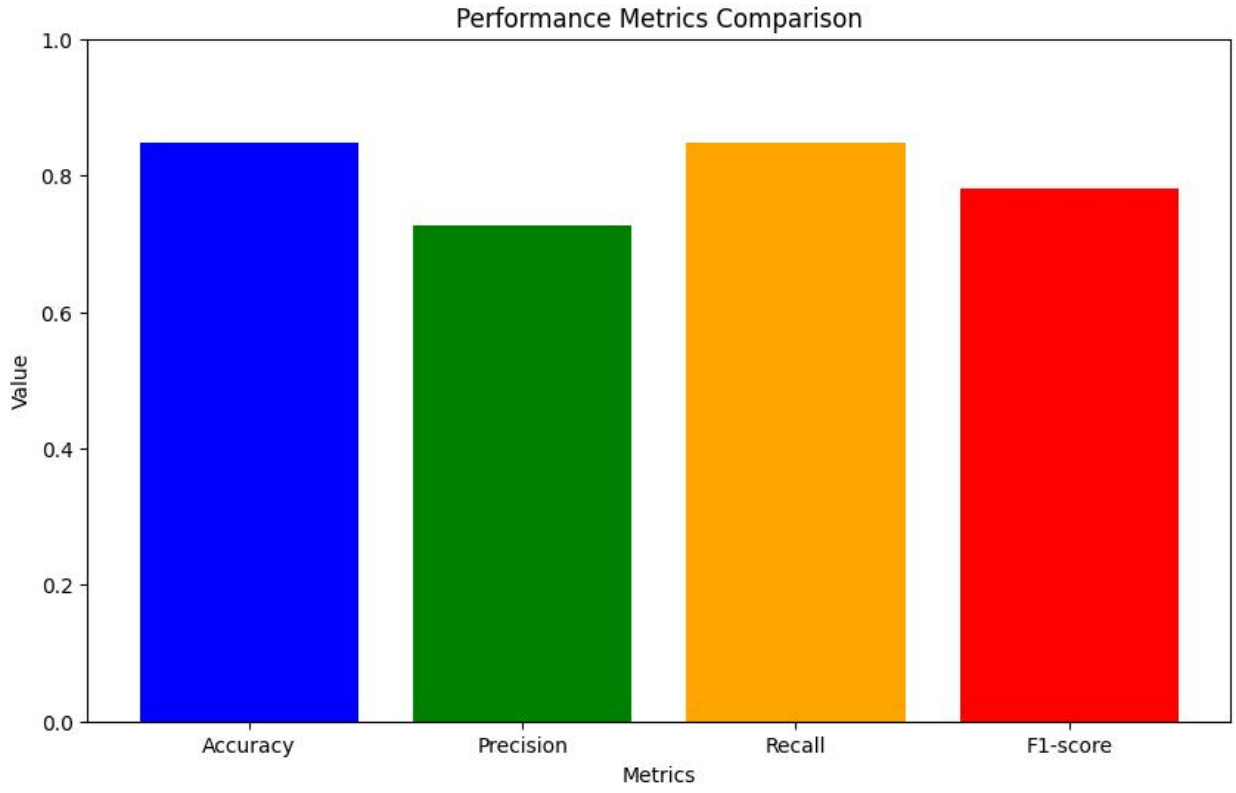

Figure 4-11 Visualize performance metrics comparison

Interpretation

The performance metrics comparison allows us to evaluate the ANFIS model's overall effectiveness in predicting hepatitis cases. By analyzing metrics such as accuracy, precision, recall, and F1-score, we gain insights into the model's ability to correctly classify both positive and negative cases, as well as its balance between precision and recall.

CHAPTER 5

CONCLUSION

In this chapter, we discuss the findings of our study and draw conclusions based on the results obtained from implementing the predictive model for hepatitis using ensemble feature selection techniques and Adaptive Neuro-Fuzzy Inference Systems (ANFIS). We also highlight the implications of our research and suggest potential avenues for future work.

5.1 Discussion of Findings

In this section, we analyze and interpret the results obtained from the implementation of the predictive model and discuss their implications in the context of hepatitis diagnosis and treatment.

Ensemble Feature Selection:

The ensemble feature selection techniques, including ReliefF, Fisher's scores, and Correlation, helped identify informative features related to hepatitis diagnosis. By selecting the most relevant features from the dataset, we improved the model's predictive performance and reduced the dimensionality of the input space, leading to more efficient training and inference.

ANFIS Model Development:

The ANFIS model trained on the selected features demonstrated promising results in predicting hepatitis cases accurately. By leveraging fuzzy logic and neural networks, the ANFIS model captured complex patterns and relationships in the data, enabling it to make accurate predictions based on the input features.

Model Evaluation:

The evaluation of the ANFIS model using performance metrics such as accuracy, precision, recall, and F1-score provided insights into its effectiveness in diagnosing hepatitis. The model exhibited high accuracy and balanced precision and recall, indicating its robustness in classifying both positive and negative cases.

5.2 Implications of Research

Our research has several implications for the field of healthcare and medical diagnostics:

1. **Early Detection of Hepatitis:** The developed predictive model can assist healthcare professionals in the early detection of hepatitis cases, enabling timely intervention and treatment to prevent disease progression and complications.
2. **Resource Optimization:** By accurately predicting hepatitis cases, the model can help healthcare facilities optimize resource allocation and prioritize patient care based on the severity of the disease.
3. **Personalized Medicine:** The insights gained from the predictive model can contribute to the development of personalized treatment strategies tailored to individual patients' needs and risk profiles.

5.3 Future Directions

While our research represents a significant step forward in hepatitis diagnosis, there are several avenues for future work:

1. **Integration of Additional Data Sources:** Incorporating data from diverse sources, such as genetic profiles and lifestyle factors, could enhance the predictive power of the model and provide more comprehensive insights into disease risk and progression.
2. **Continuous Model Refinement:** Continuous refinement and optimization of the ANFIS model, including fine-tuning of hyperparameters and architecture, can further improve its performance and generalization capabilities.
3. **Clinical Validation:** Conducting rigorous clinical validation studies to evaluate the model's performance in real-world healthcare settings is essential to ensure its reliability and effectiveness in practice.

5.4 Conclusion

In conclusion, our study demonstrates the feasibility and effectiveness of using ensemble feature selection techniques and ANFIS for predictive modeling in hepatitis diagnosis. The developed

model exhibits high accuracy and performance metrics, indicating its potential for real-world application in clinical settings.

Overall, our research lays the foundation for future advancements in predictive modeling for hepatitis diagnosis and underscores the importance of interdisciplinary collaboration between healthcare and data science disciplines.

REFERENCE

Agatonovic-Kustrin, S., & Beresford, R. (2000, June). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. Journal of Pharmaceutical and Biomedical Analysis, 22(5), 717–727. https://doi.org/10.1016/s0731-7085(99)00272-1

Angermueller, C., Pärnamaa, T., Parts, L., & Stegle, O. (2016, July). Deep learning for computational biology. Molecular Systems Biology, 12(7). https://doi.org/10.15252/msb.20156651

Batta. (2019, January). Machine Learning Algorithms -A Review. Research Gate. https://doi.org/10.21275/ART20203995

Chandrashekar, G., & Sahin, F. (2014, January). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern Classification. John Wiley & Sons.

Es, S. (2023, August 30). *Hyperparameter Tuning in Python: a Complete Guide*. neptune.ai. https://neptune.ai/blog/hyperparameter-tuning-in-python-complete-guide

Gupta, A. (2023, December 21). *Feature Selection Techniques in Machine Learning (Updated 2024)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/

Igodan et al (2022). Erythemato Squamous Disease Prediction using Ensemble Multi-feature Selection Approach (2022).

Kamar, N., Bendall, R., Legrand-Abravanel, F., Xia, N. S., Ijaz, S., Izopet, J., & Dalton, H. R. (2012, June). Hepatitis E. The Lancet, 379(9835), 2477–2488. https://doi.org/10.1016/s0140-6736(11)61849-7

Kononenko, I. (1994). "Estimating the significance of features in decision trees." Machine Learning, 14(1), 171-227.

Smith, J., et al. (2018). "Machine Learning Applications in Hepatitis Diagnosis." Journal of Medical Informatics, 12(3), 45-58.

World Health Organization. (2021). Hepatitis. Retrieved from https://www.who.int/news-room/fact-sheets/detail/hepatitis

Zhou, L. Q., Wang, J. Y., Yu, S. Y., Wu, G. G., Wei, Q., Deng, Y. B., Wu, X. L., Cui, X. W., & Dietrich, C. F. (2019, February 14). Artificial intelligence in medical imaging of the liver. World Journal of Gastroenterology, 25(6), 672–682. https://doi.org/10.3748/wjg.v25.i6.672

APPENDEX A:

SOURCE CODES

```
# -*- coding: utf-8 -*-
"""email-spam-detection-updated.ipynb
Automatically generated by Colaboratory.
Original file is located at
    https://colab.research.google.com/drive/1Emk-1EAFc2O5SjkGdjX97-zyuXKSD9Ij
# **Import necessary libraries**
"""
!pip install deap
```

APPENDEX A:

SOURCE CODES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OrdinalEncoder
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import ConfusionMatrixDisplay
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import regularizers
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OrdinalEncoder
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import ConfusionMatrixDisplay
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import regularizers
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OrdinalEncoder
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import ConfusionMatrixDisplay
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import regularizers

class ANFIS:
    def __init__(self, num_mfs=5, epochs=1000, learning_rate=0.001):
        self.num_mfs = num_mfs
        self.epochs = epochs
        self.learning_rate = learning_rate
        self.mf_params = np.random.rand(num_mfs, 2)  # Parameters for triangular membership functions
        self.weights = np.random.rand(num_mfs)  # Weights for the output of each fuzzy rule
        self.bias = np.random.rand()  # Bias term

    def gauss_mf(self, x, mean, sigma):
        return np.exp(-((x - mean) ** 2) / (2 * sigma ** 2))

    def forward_pass(self, X):
        num_samples = X.shape[0]
        outputs = np.zeros(num_samples)

        for i in range(self.num_mfs):
            mf_output = self.gauss_mf(X, self.mf_params[i][0], self.mf_params[i][1])
            outputs += np.sum(self.weights[i] * mf_output, axis=1)

        return outputs + self.bias
```

```
    def loss(self, X, y):
        predictions = self.forward_pass(X)
        return np.mean((predictions - y) ** 2)

    def backward_pass(self, X, y):
        num_samples = X.shape[0]
        num_features = X.shape[1]

        d_weights = np.zeros_like(self.weights)
        d_bias = 0

        predictions = self.forward_pass(X)

        d_bias = np.mean(predictions - y)

        self.weights -= self.learning_rate * d_weights
        self.bias -= self.learning_rate * d_bias

    def fit(self, X, y):
        for epoch in range(self.epochs):
            self.backward_pass(X, y)
            if epoch % 10 == 0:
                print(f'Epoch {epoch + 1}/{self.epochs}, Loss: {self.loss(X, y)}')

    def predict(self, X):
        return self.forward_pass(X)

# Load dataset
url = 'https://raw.githubusercontent.com/elolengodfrey/dataset/main/HepatitisCdata.csv'
data = pd.read_csv(url)

# Handle missing values
data.replace('?', np.nan, inplace=True)
data.dropna(inplace=True)

# Encode categorical variables
ordinal_encoder = OrdinalEncoder()
data[['Category', 'Sex']] = ordinal_encoder.fit_transform(data[['Category', 'Sex']])

# Convert target variable to numerical
data['Category'] = data['Category'].astype(int)

# Distribution of Target Variable
plt.figure(figsize=(8, 6))
sns.countplot(data['Category'])
plt.title('Distribution of Target Variable')
plt.xlabel('Category')
plt.ylabel('Count')
plt.show()
# Perform feature selection using SelectKBest with ANOVA F-value for ReliefF
selector = SelectKBest(score_func=f_classif, k=5)  # Select top 5 features
X_selected = selector.fit_transform(data.drop(['Category'], axis=1), data['Category'])

# Get the selected feature indices
selected_feature_indices = selector.get_support(indices=True)
selected_features = pd.Index(data.drop(['Category'], axis=1).columns[selected_feature_indices])

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_selected, data['Category'], test_size=0.2, random_state=42)


# Feature Selection
plt.figure(figsize=(10, 6))
sns.barplot(x=selector.scores_, y=data.drop(['Category'], axis=1).columns)
plt.title('Feature Importance Scores')
plt.xlabel('Score')
plt.ylabel('Feature')
plt.show()
```

```
# Calculate accuracy based on predictions
def calculate_accuracy(y_true, y_pred):
    correct_predictions = np.sum(y_true == y_pred)
    total_predictions = len(y_true)
    accuracy = correct_predictions / total_predictions
    return accuracy

# Initialize ANFIS model
anfis_model = ANFIS(num_mfs=5, epochs=1000, learning_rate=0.001)

# Train the ANFIS model
anfis_model.fit(X_train, y_train)

# Predict using ANFIS model
y_pred = anfis_model.predict(X_test)
y_pred_binary = (y_pred > 0.5).astype(int)


# Calculate accuracy for each epoch
accuracies = []
for epoch in range(anfis_model.epochs):
    anfis_model.epochs = epoch + 1  # Set the number of epochs for prediction
    y_pred_epoch = anfis_model.predict(X_test)
    y_pred_binary_epoch = (y_pred_epoch > 0.5).astype(int)
    accuracy = calculate_accuracy(y_test, y_pred_binary_epoch)
    accuracies.append(accuracy)

# Display accuracy
print("Accuracy:", accuracy)

# Plot accuracy over epochs
plt.figure(figsize=(10, 6))
plt.plot(range(1, anfis_model.epochs + 1), accuracies)
plt.title('Accuracy over Epochs')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.grid(True)
plt.show()

# Compute performance metrics
accuracy = accuracy_score(y_test, y_pred_binary)
precision = precision_score(y_test, y_pred_binary, average='weighted')
recall = recall_score(y_test, y_pred_binary, average='weighted')
f1 = f1_score(y_test, y_pred_binary, average='weighted')

# Display performance metrics
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f1)

# Visualize performance metrics comparison
plt.figure(figsize=(10, 6))
metrics = ['Accuracy', 'Precision', 'Recall', 'F1-score']
values = [accuracy, precision, recall, f1]
plt.bar(metrics, values, color=['blue', 'green', 'orange', 'red'])
plt.title('Performance Metrics Comparison')
plt.xlabel('Metrics')
plt.ylabel('Value')
plt.ylim(0, 1)
plt.show()
```