A

Major Project Report

On

A HYBRID DIABETIC FOOT ULCER RISK ASSESSMENT AND SURVIVAL PROBABILITY ESTIMATION

Submitted to

In Partial Fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

IN

AIRTIFICIAL INTELLIGENCE AND DATA SCIENCE

Submitted By

B. ARCHANA	**(218R1A7211)**
B. VAISHNAVI	**(218R1A7212)**
YAMINI SREE	**(218R1A7248)**
M. ABHISHEK	**(218R1A7201)**

Under the Esteemed guidance of

Mr. R. SRIKANTH

Assistant Professor

ENGINEERING COLLEGE

EXPLORE TO INVENT

Department of Artificial Intelligence and Data Science

CMR ENGINEERING COLLEGE UGC AUTONOMOUS

(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)

Kandlakoya, Medchal-Malkajgiri District. Hyderabad-501 401.

2024-2025

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS

(Accredited by NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad-501 401

Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the project entitled **"A Hybrid Approach for Diabetic Foot Ulcer Risk Assessment and Survival Probability Estimation"** is a Bonafide work carried out by

B. ARCHANA	**(218R1A7211)**
B. VAISHNAVI	**(218R1A7212)**
YAMINI SREE	**(218R1A7248)**
M. ABHISHEK	**(218R1A7201)**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE** from CMR Engineering College affiliated to JNTU, Hyderabad, under our guidance and supervision. The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	**Project Coordinator**	**Head of the Department**	**External Examiner**
Mr. R. Srikanth	**Mr. R. Srikanth**	**Dr. M. Laxmaiah**	
Assistant Professor	Assistant Professor	Professor & HOD	
AI&DS, CMREC	AI&DS, CMREC	AI&DS, CMREC	

DECLARATION

This is to certify that the work reported in the present project entitled **"A Hybrid Approach for Diabetic Foot Ulcer Risk Assessment and Survival Probability Estimation"** is a record of Bonafide work done by us in the Department of Artificial Intelligence and Data Science CMR Engineering College, JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

B. ARCHANA (218R1A7211)
B. VAISHNAVI (218R1A7212)
YAMINI SREE (218R1A7248)
M. ABHISHEK (218R1A7201)

ACKNOWLEDGMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. M. Laxmaiah**, HOD, **Artificial Intelligence and Data Science, CMR Engineering College** for their constant support**.**

I am extremely thankful to Mr. R Srikanth**,** Assistant Professor, Internal Guide, Artificial Intelligence and Data Science, for his/ her constant guidance, encouragement and moral support throughout the project.

I will be failing in duty if I do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

I thank **Mr. R. Srikanth,** Assistant Professor Major Project Coordinator for his constant support in carrying out the project activities and reviews.

I express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, I am very much thankful to my parents who guided me for every step.

B. ARCHANA (218R1A7211)

B. VAISHNAVI (218R1A7212)

YAMINI SREE (218R1A7248)

M. ABHISHEK (218R1A7201)

CONTENTS

LIST OF FIGURES

ABSTRACT

Diabetic Foot Ulcers (DFUs) are a severe complication of diabetes that can lead to infection, amputation, and mortality if not identified and treated in time. The increasing prevalence of diabetes worldwide highlights the need for early detection and personalized risk assessment to mitigate these consequences. Existing models often fail to account for the complex interactions between multiple high-impact factors such as Peripheral Neuropathy, Peripheral Arterial Disease (PAD), and Glycemic Control, resulting in suboptimal prediction accuracy. Moreover, these models lack survival analysis, which is crucial in estimating the time until the occurrence or recurrence of foot ulcers. To address these challenges, this research proposes a Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM) that combines the strengths of three machine learning techniques: Random Forest (RF), Deep Neural Network (DNN), and Random Survival Forest (RSF). The RF model is utilized for feature-based classification, while the DNN processes high-dimensional data, capturing intricate patterns that linear models often miss. RSF is incorporated to model time-to-event data and estimate survival probabilities, offering insights into patient-specific risks over time. An adaptive fusion mechanism is implemented to intelligently combine the logit outputs of these models and further refine the predictions through logistic regression. The fusion mechanism dynamically assigns optimal weights to the models based on their individual performance, ensuring robust and accurate predictions. The proposed system delivers not only a binary classification of DFU risk but also provides survival probabilities for different time frames (1 year, 3 years, and 5 years), offering clinicians a more comprehensive view of patient outcomes. Extensive experimentation and evaluation on a dataset demonstrate the superior performance of the HDFUPM model over traditional approaches. Key performance metrics such as accuracy, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) validate the effectiveness of the proposed system. This hybrid approach significantly improves risk prediction and survival analysis, contributing to timely interventions and better patient outcomes.

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

Diabetes is a chronic metabolic disorder that affects millions of people globally and often leads to complications such as cardiovascular diseases, kidney failure, and neuropathy. Among these complications, Diabetic Foot Ulcers (DFUs) remain one of the most debilitating and life- threatening consequences. DFUs arise due to a combination of neuropathy, ischemia, and infection, ultimately leading to tissue breakdown and ulceration. Without timely intervention, DFUs can escalate to infections that necessitate lower-limb amputation, significantly reducing the quality of life and increasing mortality rates.

Despite advancements in medical technology, the accurate prediction of DFU risk remains a challenge. This is because DFU development is influenced by a wide range of factors, including Peripheral Neuropathy, Peripheral Arterial Disease (PAD), Poor Glycemic Control, Foot Deformities, and Previous History of Ulcers. Traditional prediction models often fail to capture the complex interdependencies between these factors, leading to unreliable predictions. Moreover, they do not account for the time-to-event component, which is essential in estimating the likelihood of ulcer development over time.

1.2 PROBLEM STATEMENT

Existing DFU prediction models primarily focus on binary classification, overlooking the temporal aspect of disease progression. These models exhibit limited ability to capture the multifaceted relationship between clinical parameters and ulcer formation. Furthermore, they often neglect the estimation of survival probabilities, which is vital for understanding patient-specific risks over different time frames. The absence of an integrated approach that combines classification and survival analysis results in incomplete insights, reducing the effectiveness of preventive strategies and personalized patient care.

The primary objective of this research is to develop a Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM) that leverages multiple machine learning techniques to provide accurate risk assessment and survival analysis. The key objectives are: To design a hybrid model that integrates Random Forest, Deep Neural Network, and Random Survival Forest for improved DFU prediction.

To implement an adaptive fusion mechanism that dynamically combines the outputs of the three models and optimizes the final prediction using logistic regression. To incorporate survival analysis using RSF, enabling the estimation of survival probabilities at different time intervals (1 year, 3 years, and 5 years). To evaluate the proposed model's performance using key metrics such as accuracy, AUC-ROC, and F1-score, ensuring that the hybrid model outperforms existing prediction techniques. The scope of this project extends to the design, implementation, and evaluation of a hybrid model for DFU risk prediction and survival analysis. The model considers a wide range of clinical and self- assessment factors to deliver personalized risk scores and survival probabilities. By leveraging multiple machine learning techniques and combining their strengths through adaptive fusion, the proposed system provides an enhanced level of accuracy and robustness. Additionally, the project aims to facilitate improved decision-making for clinicians by offering a comprehensive view of patient risk trajectories.

CHAPTER 2
LITERATURE SURVEY

Diabetes and its associated complications, particularly Diabetic Foot Ulcers (DFUs), continue to be a major public health challenge, contributing to high mortality rates and imposing significant financial burdens on healthcare systems worldwide. Diabetes has reached pandemic proportions, necessitating comprehensive prevention and management strategies to mitigate its consequences. Studies emphasize the global mortality associated with DFUs, underscoring the critical need for early detection and timely intervention. Identifying at-risk patients through multi- intervention approaches has shown significant promise in reducing the likelihood of ulceration, particularly in individuals with type 2 diabetes mellitus. Furthermore, understanding the pathophysiology of foot ulcers highlights the importance of interdisciplinary collaboration to ensure effective management and improved outcomes.

With the advancement of artificial intelligence (AI) and deep learning technologies, innovative approaches have emerged for medical diagnostics and disease management. AI models have demonstrated high accuracy in disease classification and risk prediction, setting new benchmarks for AI applications in healthcare. Similarly, AI has been leveraged to address diabetic complications, with automated systems improving diagnostic accuracy and accessibility. The integration of AI into imaging and diagnostic systems across various medical domains further underscores its transformative potential in enhancing healthcare outcomes.

Despite these technological advances, the application of AI in healthcare raises significant ethical and governance concerns. Guidelines emphasize the responsible use of AI in healthcare, focusing on protecting patient data and ensuring equitable access to AI-driven technologies. The multidimensional nature of these concerns highlights the need for effective frameworks to manage information security and ethical challenges in AI applications. Moreover, continuous innovation in diabetes management, coupled with the growing acceptance of AI-powered solutions, is crucial for enhancing patient care and outcomes.

In the context of diabetic foot ulcer prediction, the integration of AI models such as Random Forest (RF), Deep Neural Networks (DNN), and Random Survival Forest (RSF) has shown immense potential in improving predictive accuracy and survival analysis. These models not only enhance the early identification of high-risk patients but also provide actionable insights for preventive care, thereby reducing the overall incidence and severity of diabetic foot ulcers. The hybrid approach adopted in the Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM) leverages the strengths of these models to deliver robust and reliable predictions while addressing the limitations of traditional methodologies. This literature review provides a comprehensive understanding of the current state of DFU prediction, highlighting the need for an integrated approach that combines AI-driven classification models with survival analysis techniques to improve the accuracy of risk assessment and enhance patient outcomes.

CHAPTER 3
EXISTING SYSTEM

3.1 MACHINE LEARNING MODELS FOR DFU PREDICTION

The existing systems used for Diabetic Foot Ulcer (DFU) risk prediction primarily rely on machine learning techniques such as Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM), and Decision Trees. These models analyze patient data, including clinical parameters and self-assessment inputs, to classify individuals into high-risk and low-risk categories. However, these models primarily provide binary classification and do not account for the temporal aspects of DFU progression.

Logistic Regression (LR) is one of the simplest models used for classification, but it performs poorly with high-dimensional data and complex feature interactions. Decision Trees and Random Forest (RF) have demonstrated better predictive accuracy due to their ability to handle non-linear relationships and multiple features, making them suitable for DFU prediction. However, they are limited to static classification and do not model time-dependent outcomes.

3.2 DEEP LEARNING APPROACHES

Deep learning models, particularly Deep Neural Networks (DNNs), have been employed to enhance predictive accuracy by capturing complex patterns in high-dimensional data. DNNs process large amounts of clinical and demographic data to learn intricate relationships between input variables. However, these models require extensive training data, and their predictions lack interpretability, making them less favorable for clinical decision-making. Furthermore, DNNs, like other classification models, do not provide insights into the time frame in which DFUs are likely to develop, making them inadequate for survival analysis.

3.3 SURVIVAL ANALYSIS TECHNIQUES

Survival analysis models, such as Kaplan-Meier Estimator and Cox Proportional Hazards Model, have been utilized to estimate time-to-event outcomes in medical research. However, these models rely on strict assumptions about proportional hazards and may not perform well with high-dimensional data. Random Survival Forest (RSF) has emerged as a more flexible alternative that extends the capabilities of traditional survival models by handling complex feature relationships and providing survival probability estimates.

3.4 LIMITATIONS OF EXISTING SYSTEMS

1. Lack of Integration of Multiple Models: Current approaches do not combine classification models with survival analysis techniques, leading to fragmented predictions.
2. Absence of Time-to-Event Predictions: Most models provide only binary classification outcomes without accounting for the time frame within which DFUs may develop.
3. Limited Model Fusion Capabilities: Existing systems do not utilize adaptive fusion mechanisms to dynamically assign optimal weights to different models, resulting in suboptimal performance.
4. These limitations highlight the need for an integrated hybrid approach that combines classification and survival analysis while leveraging adaptive fusion to refine predictions. The proposed Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM) addresses these gaps by integrating RF, DNN, and RSF, providing comprehensive risk assessment and survival probability estimates.

CHAPTER 4
PROPOSED SYSTEM

4.1 HYBRID DIABETIC FOOT ULCER PREDICTION MODEL (HDFUPM)

The proposed Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM) aims to overcome the limitations of existing systems by integrating classification and survival analysis models through an adaptive fusion mechanism. This hybrid approach combines the strengths of Random Forest (RF) for classification, Deep Neural Network (DNN) for feature extraction and pattern recognition, and Random Survival Forest (RSF) for time-to-event prediction, thereby providing a comprehensive assessment of DFU risk along with survival probability estimates over time.

4.2 MODEL COMPONENTS

The model comprises of three components for two modules that include risk assessment module and survival analysis module.

1. Random Forest (RF): RF serves as the primary classification model, leveraging an ensemble of decision trees to predict whether a patient is at high or low risk of developing DFUs. It processes static clinical data and self-assessment responses to generate initial classification outputs.
2. Deep Neural Network (DNN): The DNN model is employed to capture complex, non- linear relationships between patient features. It processes high-dimensional input data, extracts relevant patterns, and enhances classification performance. The DNN output contributes to the final risk score by improving predictive accuracy.
3. Random Survival Forest (RSF): RSF estimates survival probabilities by modeling the time- to-event data associated with DFU onset or recurrence. It evaluates the likelihood of a patient developing DFUs over short-term, medium-term, and long-term periods, offering insights into the temporal dimension of risk.

4.3 ADAPTIVE FUSION MECHANISM

The adaptive fusion mechanism is a critical component in the Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM) that dynamically combines the outputs from the Random Forest (RF), Deep Neural Network (DNN), and Random Survival Forest (RSF) models to generate a comprehensive risk score. Since RF handles tabular data effectively, DNN captures complex non- linear relationships, and RSF models time-to-event outcomes, the fusion mechanism ensures that the combined prediction leverages the strengths of all three models. This approach improves overall prediction accuracy and enhances the system's ability to assess both the likelihood of developing DFUs and the survival probability over specified time periods.

The fusion process begins by normalizing the outputs from the three models onto a uniform scale. RF and DNN provide classification probabilities, while RSF generates survival probabilities, which are inverted to reflect the risk. The normalized values are then combined through a weighted sum, where each model's output is assigned a dynamic weight based on its performance and reliability. The final combined output is passed through a sigmoid function to produce a probability score between 0 and 1, reflecting the patient's risk level.

A key advantage of this adaptive mechanism is its ability to dynamically adjust the contribution of each model based on real-time feedback and the complexity of the input data. This adaptability makes the fusion process robust, ensuring that the model performs optimally across diverse patient profiles. Furthermore, by combining the predictive capabilities of multiple models, the system minimizes the limitations inherent in individual approaches and provides clinicians with more reliable and interpretable risk assessments.

The adaptive fusion mechanism enhances not only the prediction of DFU risk but also survival analysis over different timeframes (1 year, 3 years, and 5 years). This integrated approach ensures that the system generates more precise and clinically relevant outcomes, supporting informed decision- making and personalized patient management.

4.4 RISK CLASSIFICATION AND SURVIVAL ANALYSIS

This model has two main modules that calculates risk of developing DFU and survival probabilities of each patient.

4.4.1 Risk Classification Module:

The risk classification module forms the core of the Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM) by assessing the likelihood of developing diabetic foot ulcers (DFUs) in patients. It integrates two distinct machine learning models: Random Forest (RF) and Deep Neural Network (DNN), each contributing unique strengths to the predictive process. RF, being an ensemble-based model, efficiently handles structured tabular data and provides high interpretability, making it well-suited for analyzing clinical data and self-assessment responses. It leverages decision trees to perform feature splits and assigns probabilistic scores to classify patients into high-risk or low-risk categories. On the other hand, DNN introduces the ability to capture complex non-linear patterns within the data, especially when feature interactions are intricate and not easily captured by traditional models. It uses multiple hidden layers and activation functions to identify subtle relationships between various risk factors, such as duration of diabetes, HbA1c levels, foot deformities, and peripheral neuropathy. DNN is particularly effective when dealing with large datasets and unstructured information, enabling the system to learn deep representations that contribute to improving prediction accuracy.

To ensure robustness and improve the overall classification outcome, the adaptive fusion mechanism integrates the outputs of both RF and DNN models. After normalization, the combined predictions from these models are weighted dynamically based on their respective performance and reliability. This process generates a comprehensive risk score that reflects the probability of a patient developing DFUs, providing clinicians with a more accurate and nuanced assessment for personalized management.

4.4.2 Survival Analysis Module:

In addition to predicting DFU risk, the survival analysis module focuses on modeling the time-to-event outcomes by predicting the likelihood of survival over specified time periods. This is achieved through the implementation of the Random Survival Forest (RSF), which extends the concept of Random Forest to survival data by estimating the hazard function and cumulative survival probabilities. RSF is well-suited for time-to-event data as it accounts for censoring, ensuring that incomplete observations do not bias the results.

RSF constructs multiple survival trees where each tree partitions the data based on features that influence survival time, such as age, PAD status, glycemic control, and history of prior DFUs. The ensemble of survival trees then produces cumulative hazard and survival estimates for each patient. By averaging the survival probabilities from all the trees, RSF provides a reliable estimation of the patient's likelihood of remaining ulcer-free over different time frames — typically 1 year, 3 years, and 5 years.

The integration of survival analysis with risk classification ensures that the model not only identifies high-risk patients but also predicts the timeframe within which they are most likely to develop DFUs. This dual capability enhances the model’s clinical relevance by enabling timely interventions and personalized treatment plans. Clinicians can leverage these insights to prioritize patients who require immediate attention and allocate resources more effectively.

The combined functionality of risk classification and survival analysis, augmented by the adaptive fusion mechanism, positions the HDFUPM as a comprehensive and reliable predictive framework for DFU prevention and long-term patient care.

4.5 ADVANTAGES OF THE PROPOSED SYSTEM

The Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM) is a offers several advantages over traditional models and existing prediction frameworks by combining machine learning-based risk classification, survival analysis, and an adaptive fusion mechanism. This multifaceted approach enhances the system's accuracy, reliability, and clinical applicability, ensuring better outcomes for diabetic patients at risk of developing foot ulcers.

1. Higher Prediction Accuracy

   The integration of Random Forest (RF) and Deep Neural Network (DNN) ensures that both linear and non-linear relationships among features are captured effectively. RF handles structured tabular data efficiently, while DNN excels at capturing complex interactions and subtle patterns. The adaptive fusion mechanism combines the strengths of both models, dynamically adjusting their contributions to minimize classification errors and improve overall predictive performance.

2. Robust Time-to-Event Modeling

   The inclusion of Random Survival Forest (RSF) enhances the system by predicting not only the risk of developing diabetic foot ulcers (DFUs) but also the time-to-event outcome. RSF models survival probabilities over different time periods (1 year, 3 years, and 5 years), allowing clinicians to estimate the likelihood of ulcer-free survival for individual patients. This time-sensitive analysis helps prioritize high-risk patients and ensures timely preventive interventions.

3. Enhanced Clinical Interpretation and Trustworthiness

   The combination of interpretable models such as RF and RSF with the accuracy of DNN improves the trustworthiness of predictions in clinical settings. RF and RSF provide interpretable insights into feature importance and survival probabilities, while DNN adds depth to the classification process. The adaptive fusion mechanism balances these elements, ensuring reliable outcomes that clinicians can confidently rely on for decision- making.

4.5 HOW HDFUPM WORKS

Let's now understand how our model works step by step:

Step-1: Data Collection & Preprocessing

Patient data is collected from publicly available datasets, self-assessment tools (questionnaires), and clinical reports.

Synthetic data is generated to balance the dataset.

The dataset undergoes preprocessing, including handling missing values, normalizing numerical features, and encoding categorical variables.

Step-2: Feature Selection & Preparation

The model considers essential features based on research studies, such as the Duration of Diabetes, HbA1c Levels, Peripheral Neuropathy, Peripheral Artery Disease, Body Mass Index, Age, Foot Deformities, and Previous History of Foot Ulcers. The dataset is split into 70% training, 15% validation, and 15% testing for model evaluation.

Step-3: Modeling with Random Forest & Deep Neural Networks

Random Forest, in this stage, the initial risk prediction is based on the Random Forest (RF) algorithm. To find important patterns linked to the risks of diabetic foot ulcers, RF was trained on the combined dataset, which consisted of both clinical and self-assessment data. RF lowers the chance of overfitting while producing reliable predictions using an ensemble of decision trees. Random search methods are used for hyper parameter tuning, which include changing the maximum depth of trees or the number of estimators.

Risk Score Calculation: Random Forest aggregates the predictions of individual decision trees.

$$P(Ck|X) = \Sigma t = I(ht(X)\,Ck)$$

Where:

P(Ck | X): Probability of class Ck for given input X. T:

Total number of trees.

ht (X): Prediction of the tth tree.

I(.): Indicator function (1 if true, 0 otherwise).

The risk score can be derived from P(Ck |X) .The logit output (raw score before applying probability transformation), inverse sigmoid of probability is also extracted.

Deep Neural Network (DNN) Model, was created to manage complex and non-linear relationships in the dataset. The multilayer-architecture contains input, hidden, and output layers. Dropout layers aid in preventing overfitting, whereas buried layers use ReLU activation functions for faster and more efficient learning. An optimizer, such as Adam, was used to train the model, and backpropagation was used to change the weights for the best results. This DNN model provides a risk score and its logit output.

To improve the model during training, the loss curves were tracked in conjunction with precision, recall, and accuracy measurements. In the case of DNN, additional regularization techniques, such as L2 regularization, were studied to further reduce overfitting. The design is fine-tuned iteratively by increasing the number of neurons in each layer and evaluating activation functions beyond ReLU, such as Leaky ReLU or ELU, to identify gains in learning efficiency.

$$Z[l] = W[l]a[l-1]b[l]$$
$$a[l] = g(z[l])$$

Step-4: Adaptive Fusion Mechanism

Logit outputs from both the Random Forest and DNN models are taken. An adaptive fusion mechanism is applied to combine predictions, enhancing robustness and accuracy.

The logit score from each model are used to compute confidence scores using sigmoid function. This confidence scores are then normalized to determine the contribution of each model.

$$C_{RF} = \frac{1}{(1 + e^{(-Z_{RF})})}$$

$$C_{DNN} = \frac{1}{(1 + e^{(-Z_{DNN})})}$$

Step-5: Final Prediction Using Logistic Regression

The fused prediction is passed through a Logistic Regression model for final risk assessment. This model refines the output by learning the best combination of fused features. The input features for logistic regression model include the probability scores, along with respective confidence scores and is trained using binary cross-entropy loss.

This hybrid approach benefits from the interpretability and robustness of the RF and adaptability of the DNN in handling complex data patterns. The combined model was evaluated using integrated metrics such as AUC-ROC and. The result is a comprehensive and robust risk prediction model tailored for diabetic foot ulcer analysis.

Furthermore, a thorough comparison between the hybrid approach and the individual models (RF and DNN) was provided, and statistical tests were performed to confirm the significance of the improvements made by the hybrid model, ensuring that the hybrid approach not only combines the strengths of both models but also achieves statistically significant enhancements in performance. Finally, the model was tested on an external validation dataset to ensure robustness in unseen data scenarios, which is essential for evaluating the model's scalability and reliability in real-world clinical settings. By combining interpretability and advanced predictive capabilities, the proposed hybrid model establishes a solid basis for addressing the risks of diabetic foot ulcers, with the goal of both clinical impact and state-of-the-art predictive modeling performance.

Step-6: Survival Probability Estimation Using Random Survival Forest (RSF)

Patients' survival probabilities for 1 year, 3 years, and 5 years are estimated using the RSF model. The survival probabilities indicate the likelihood of the patient being alive over different time frames. The risk assessment outputs serve as input to this module.

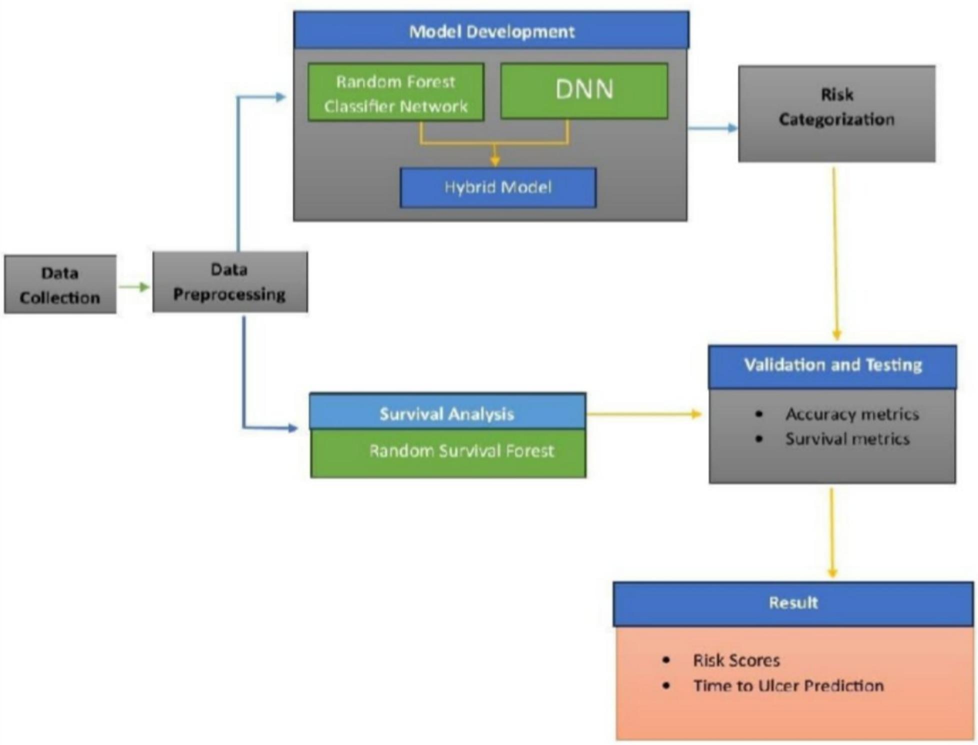

Fig.4.1 Block diagram for proposed system.

CHAPTER 5
UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The goal is for UML to become a common language for creating models of object-oriented computer software. The UML diagrams provide a comprehensive visual representation of the Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM), depicting various aspects of the system's architecture, behavior, and interactions. These diagrams are essential for understanding the design and ensuring smooth implementation.

5.1 CLASS DIAGRAM

The class diagram provides a static representation of the system by defining its key classes, attributes, and the relationships between these components. The primary classes include the User class, which represents the patient providing self-assessment data and clinical history that serves as input for the prediction model. The Risk Assessment Module class integrates the Random Forest (RF) and Deep Neural Network (DNN) models, responsible for classifying the patient's risk of developing diabetic foot ulcers. The Survival Analysis Module class employs the Random Survival Forest (RSF) model to estimate survival probabilities, enabling time-to-event analysis. The Fusion Mechanism class combines the outputs from RF, DNN, and RSF, applying an adaptive fusion mechanism to calculate the final risk score. The Database class securely stores patient data, including clinical features, self-assessment responses, and prediction results, ensuring data integrity and enabling efficient retrieval for future analysis.

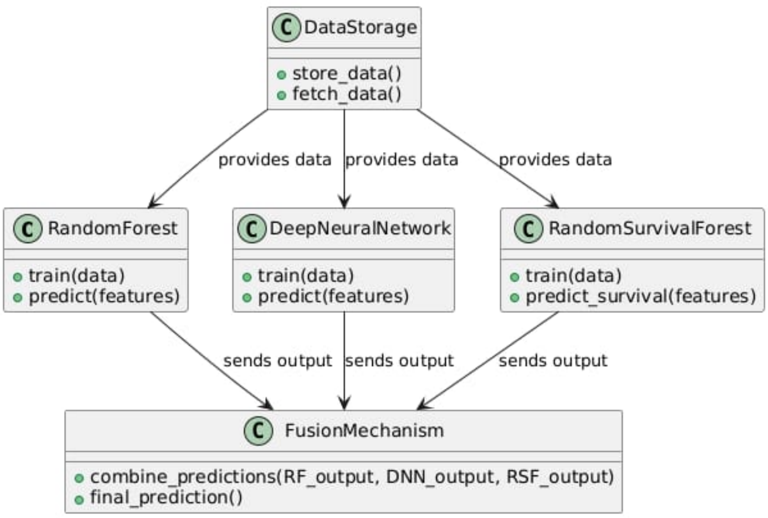

Fig.5.1 Class Diagram

5.2 USE CASE DIAGRAM

The use case diagram captures the high-level interactions between system users and various modules, providing an overview of system functionality. This information is used to generate predictions regarding DFU risk and survival probabilities. The Clinician actor accesses the risk classification results and survival probabilities to make informed decisions regarding patient care. The System processes the provided data by executing classification models, calculating survival probabilities, and generating a comprehensive risk assessment report. The interaction between these actors ensures seamless integration of input collection, prediction processing, and result interpretation.

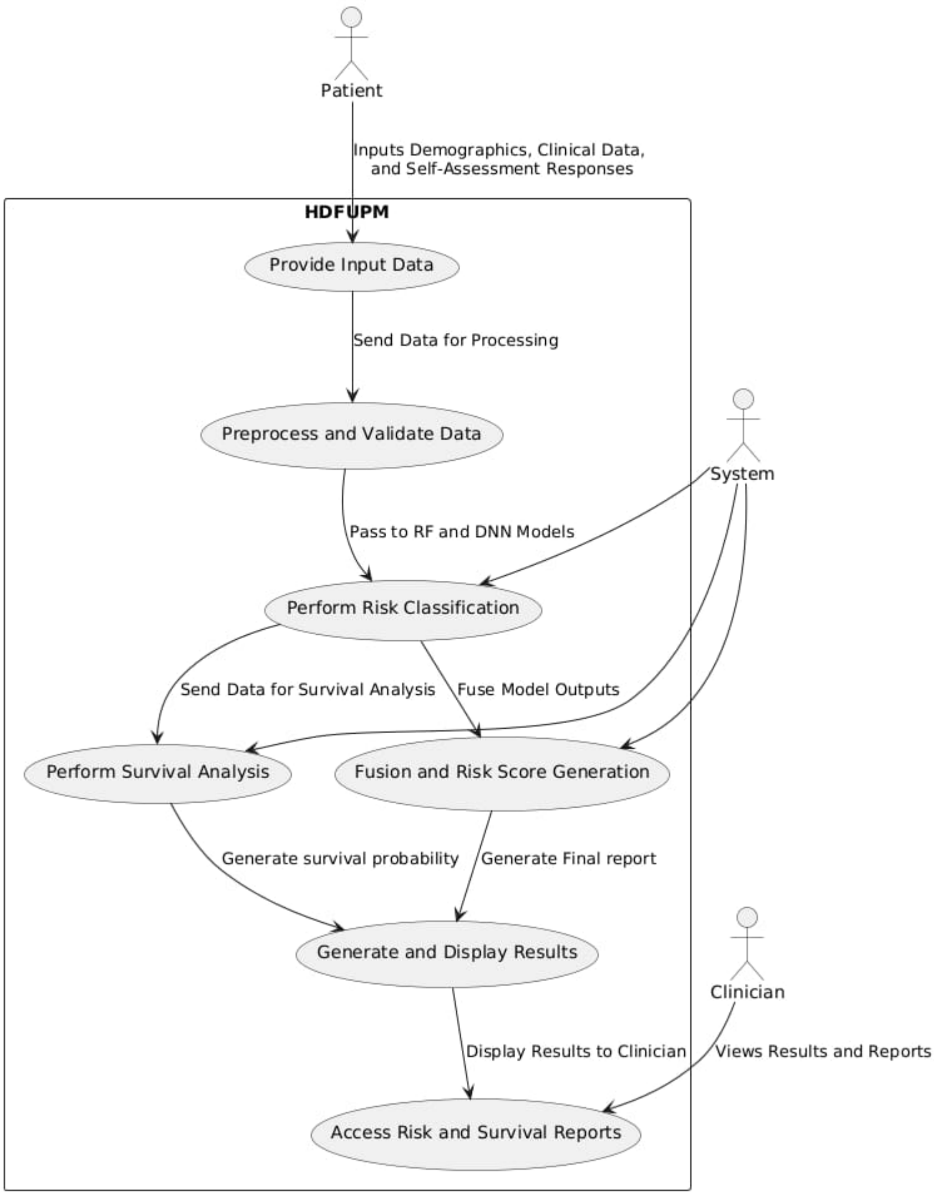

Fig.5.2 Use Case Diagram

5.3 SEQUENCE DIAGRAM

The sequence diagram models the sequential interactions between system components and their respective tasks over time. The workflow begins with the User Input step, where the patient submits their data, which is preprocessed to ensure its validity and consistency. The Data Processing step involves passing the input data through the RF and DNN models for initial predictions. In the Classification and Survival Analysis step, RF and DNN generate independent predictions while RSF estimates survival probabilities for different time intervals. These outputs are then passed to the Fusion Mechanism, where the adaptive fusion technique combines them to produce the final risk score. Finally, the Report Generation step displays the comprehensive results, including the risk assessment and survival probabilities, to the clinician for review.

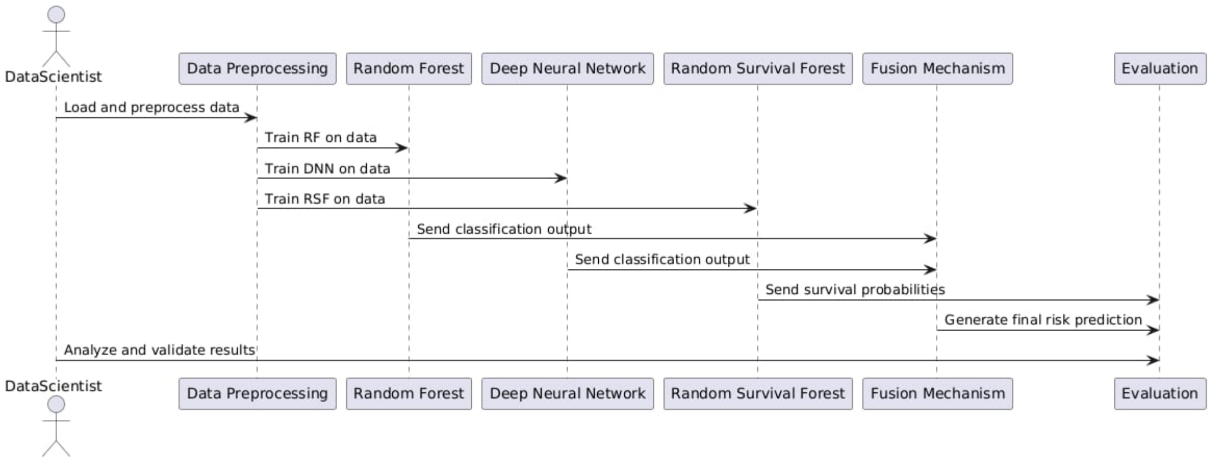

Fig.5.3 Sequence Diagram

5.4 COMPONENT DIAGRAM

The component diagram illustrates the structural relationship between various system modules that work in unison to accomplish the prediction tasks. The User Interface Module is responsible for managing data input and displaying prediction results to the user. The Prediction Engine houses the RF, DNN, and RSF models, which work collectively to assess the patient's risk and estimate survival probabilities. The Fusion Mechanism Module combines the predictions from all models and applies the adaptive fusion mechanism to derive the final risk score. The Data Storage Module securely stores patient information, clinical features, prediction results, and other relevant data, ensuring that future analysis and retrieval can be performed efficiently.

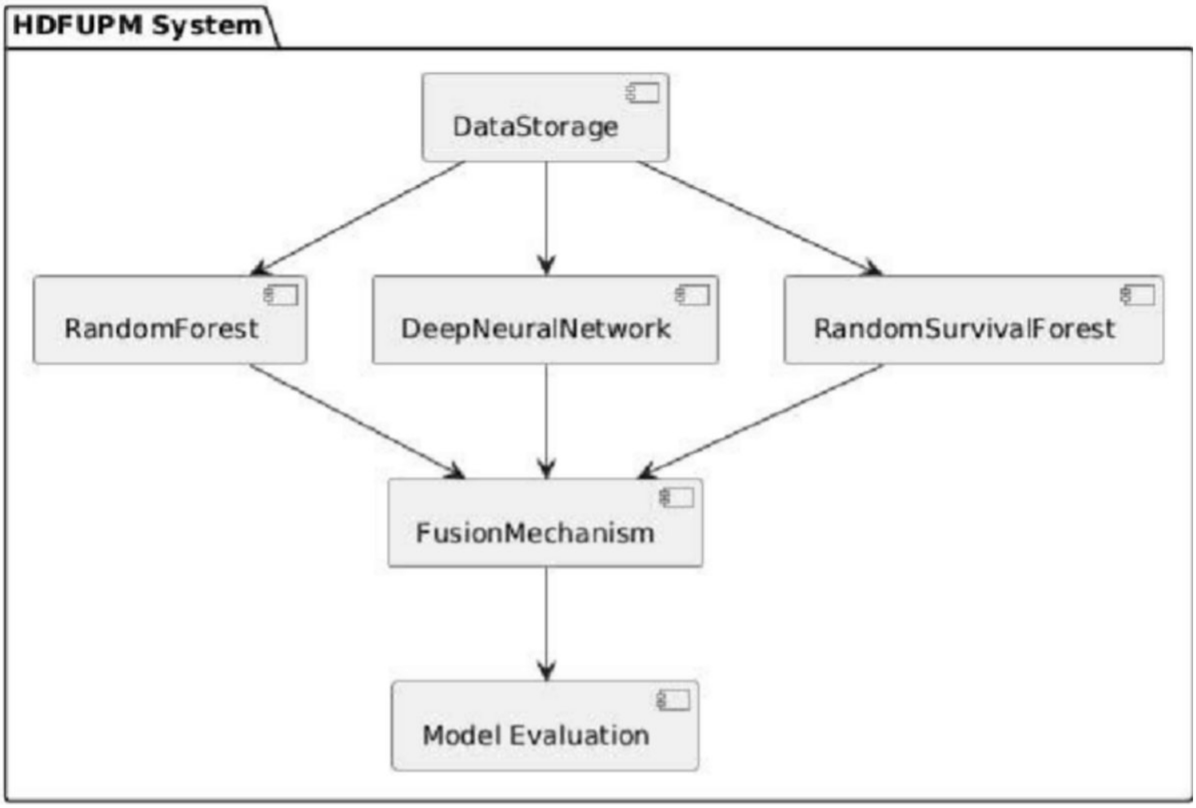

Fig 5.4 Component Diagram

5.5 ACTIVITY DIAGRAM

The activity diagram outlines the dynamic workflow within the system by mapping the sequential flow of control across various activities. The Data Input and Validation step involves collecting patient data and validating its consistency and completeness. Once validated, the Model Processing step executes classification and survival analysis using RF, DNN, and RSF models. In the Fusion and Risk Calculation step, the model outputs are combined using the adaptive fusion mechanism to compute the final risk score. Finally, the Result Display step presents the classification results and survival probabilities to the clinician, ensuring that the generated insights can be used for informed decision-making. The activity diagram represents the step-by-step flow of control within the proposed Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM). It visually outlines the dynamic behavior of the system by capturing the major activities from the initial data input to the final risk prediction. The process begins with the collection of patient data, including self-assessment responses and clinical factors such as HbA1c, neuropathy, PAD, and duration of diabetes. This is followed by data preprocessing to handle missing values and normalize inputs. The cleaned data is then simultaneously passed through the Random Forest and Deep Neural Network models, where each model computes its individual risk score.

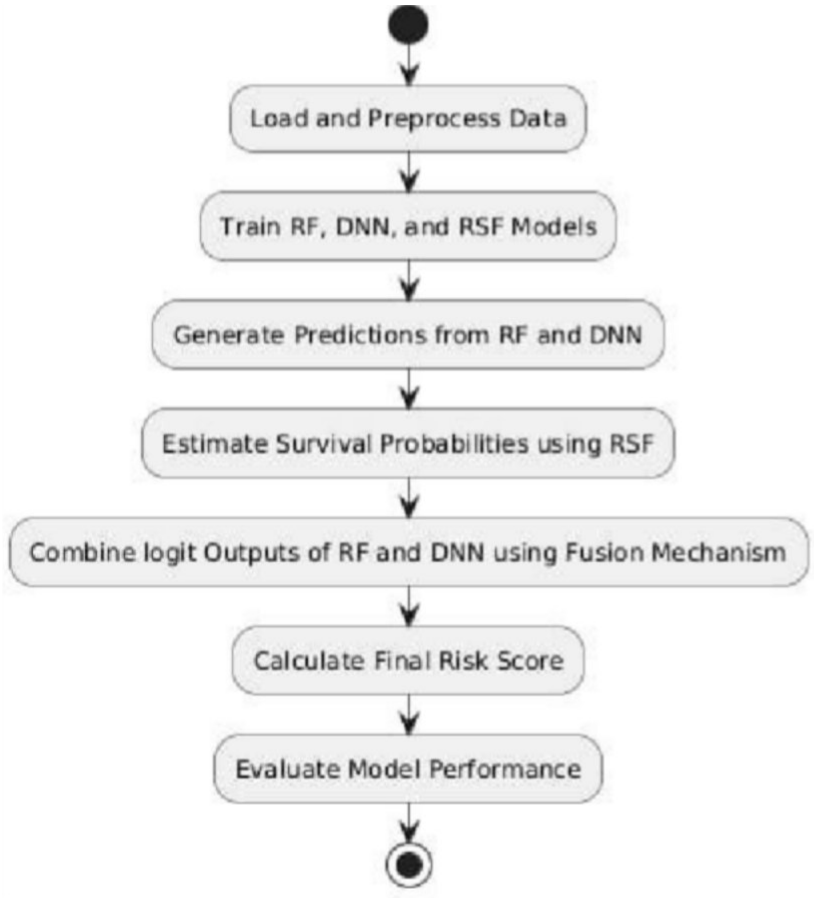

Fig.5.5 Activity Diagram

5.6 DEPLOYMENT DIAGRAM

The deployment diagram represents the physical architecture of the system, highlighting the hardware and software components involved. The Client Node hosts the interface where the patient inputs data and accesses prediction results. The Application Server processes the input data by running the classification models, executing survival analysis, and applying the fusion mechanism to generate the final prediction. The Database Server stores patient records, clinical data, and prediction results, ensuring secure storage and quick retrieval of data. This architecture ensures seamless communication between the components, enabling efficient model execution and result visualization. The deployment diagram illustrates the physical architecture and infrastructure of the proposed Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM). It shows how software components are deployed on hardware nodes to enable end-to-end system functionality. The architecture consists of client-side interfaces, where healthcare professionals or patients can input data through a web or mobile application, and a centralized backend server, which handles all the processing and computation tasks. The server hosts essential components like the data preprocessing module, machine learning models (Random Forest, Deep Neural Network, and Random Survival Forest), and the fusion mechanism with logistic regression for final predictions.

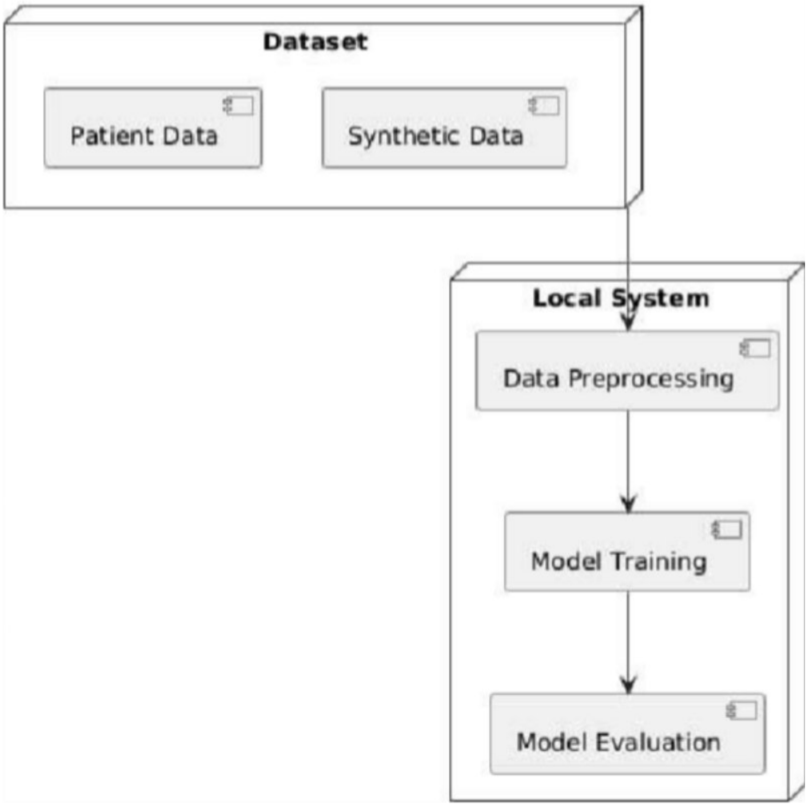

Fig.5.6 Deployment Diagram

CHAPTER 6
IMPLEMENTATION

6.1 RANDOM FOREST

Random Forest is an ensemble learning algorithm that constructs multiple decision trees during training and combines their outputs to improve accuracy and reduce overfitting. Each tree in the forest is trained on a randomly selected subset of data and features, ensuring diversity in predictions. The final prediction is obtained through majority voting (classification) or averaging (regression). RF is widely used for its robustness, interpretability, and ability to handle both numerical and categorical data.

6.1.1 Categories of Random Forest:

Random Forest can be broadly categorized based on the type of task it performs Classification and Regression. In classification tasks, Random Forest predicts the category to which a particular data point belongs by taking a majority vote of the predictions from all individual decision trees. This is particularly useful for medical applications where predicting a categorical outcome, such as the presence or absence of a condition, is required. In regression tasks, Random Forest predicts continuous values by averaging the outputs of multiple decision trees. This type of Random Forest is used in scenarios where quantitative predictions, such as survival probabilities or progression risk, are necessary. A less common variation of Random Forest includes Survival Random Forests (RSF), which models time-to-event data, making it applicable for predicting survival probabilities over different time frames.

6.1.2 Challenges of Random Forest:

While Random Forest is a powerful and flexible algorithm, it is not without challenges. One of the primary challenges is the computational complexity associated with training a large number of decision trees, which can be resource-intensive and time-consuming, especially for large datasets. Another challenge is the interpretability of the model. Although Random Forest provides feature importance scores, the ensemble nature of the model makes it difficult to trace how individual predictions are made, limiting explainability in sensitive applications such as healthcare. Furthermore, overfitting may occur when the model is trained with an excessive number of trees or when the model complexity is too high, although this is mitigated to some extent by bootstrapping and feature randomization.

6.1.3 Applications of Random Forest:

Random Forest is widely applied in various domains due to its versatility and accuracy. In healthcare, it is used for disease classification and prediction, such as predicting the risk of developing diabetic foot ulcers based on clinical and demographic data. It is also applied in genomic research to identify important genetic markers for disease prediction and classification. In finance, Random Forest is used for credit scoring, fraud detection, and risk assessment, where it analyzes complex patterns in customer behavior. Additionally, it finds applications in natural language processing (NLP) and image classification, where ensemble models improve classification accuracy. In the context of the Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM), Random Forest assists in initial risk classification, ensuring that high-risk patients are flagged for further analysis and assessment.

6.1.4 Advantages of Random Forest:

One of the primary advantages of Random Forest is its high accuracy and robustness due to its ensemble nature. By combining multiple decision trees, Random Forest reduces the risk of overfitting and improves generalization across unseen data. It also handles high-dimensional data effectively, making it suitable for complex datasets with numerous features. Another significant advantage is its ability to handle missing data by using surrogate splits, ensuring that incomplete records do not hinder model performance. Moreover, Random Forest provides feature importance scores, allowing practitioners to identify which factors contribute the most to predictions, making it easier to fine-tune and optimize the model

6.1.5 Disadvantages of Random Forest:

Despite its numerous advantages, like any other Random Forest has some limitations. Computational inefficiency is one of the primary disadvantages, especially when dealing with large datasets or when the number of trees is excessively high. This results in longer training and prediction times, which can be a challenge in real-time applications. Lack of interpretability is another drawback, as the predictions generated by an ensemble of decision trees make it difficult to explain how a specific prediction was derived. Additionally, while Random Forest reduces the risk of overfitting, it is still vulnerable to overfitting in the presence of noise or irrelevant features, particularly when the dataset is large and complex.

6.2 DEEP NEURAL NETWORKS

A Deep Neural Network (DNN) is a type of artificial neural network composed of multiple layers of interconnected neurons that process data in a hierarchical manner. It consists of an input layer that receives raw data, hidden layers that apply activation functions (such as ReLU or Sigmoid) to capture complex patterns and relationships, and an output layer that generates predictions. DNNs excel at learning non-linear features and are widely used in tasks requiring high-level feature extraction, such as image recognition, natural language processing, and medical diagnostics. Their ability to model intricate data dependencies makes them highly effective for predictive analytics in healthcare applications.

6.2.1 Categories of Deep Neural Networks:

Deep Neural Networks (DNNs) encompass a variety of architectures tailored to handle different types of data and tasks. Feedforward Neural Networks (FNNs) are the simplest and most common type of neural network where data flows in one direction from input to output. These networks consist of multiple layers, including input, hidden, and output layers, and are typically used for classification and regression tasks. Convolutional Neural Networks (CNNs), a specialized class of DNNs, are designed to process image and spatial data. CNNs employ convolutional layers that detect patterns such as edges, textures, and shapes, making them particularly effective in computer vision applications such as medical image analysis, facial recognition, and object detection.

Recurrent Neural Networks (RNNs) handle sequential data by maintaining a memory of previous inputs, allowing the model to learn temporal dependencies. However, traditional RNNs suffer from vanishing gradient problems, which limit their ability to learn long-term dependencies. To address this, advanced architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) were introduced, allowing the network to retain information over longer sequences. These models are widely used in natural language processing (NLP), time-series forecasting, and speech recognition. Autoencoders are another type of DNN used for unsupervised learning, where the network compresses the input into a latent representation and then reconstructs the original input, making them ideal for tasks like anomaly detection and dimensionality reduction. Finally, Generative Adversarial Networks (GANs) involve two competing networks — a generator and a discriminator — where the generator aims to produce synthetic data that is indistinguishable from real data, and the discriminator evaluates the generated data. GANs have shown remarkable success in generating realistic images, audio, and video data.

6.2.2 Challenges of Deep Neural Networks:

Despite their effectiveness, Deep Neural Networks come with several challenges. One of the most significant challenges is overfitting, where the model becomes overly complex and memorizes the training data instead of generalizing to unseen data. This issue is particularly prevalent in DNNs with a large number of parameters and limited training data. Techniques such as dropout, regularization, and early stopping are commonly employed to mitigate overfitting. Another major challenge is the computational intensity associated with training deep networks. DNNs require extensive computational resources, often involving Graphics Processing Units (GPUs) or Tensor Processing Units (TPUs) to accelerate training. For large datasets, training a DNN can take hours or even days, making it a resource-intensive process.

Hyperparameter tuning poses another challenge, as finding the optimal architecture, learning rate, and batch size requires a significant amount of trial and error. Poorly chosen hyperparameters can lead to underfitting or overfitting, adversely affecting model performance. Moreover, DNNs are often characterized as black-box models, making them difficult to interpret and explain. Understanding how a deep model arrives at a specific decision remains a critical concern, particularly in domains such as healthcare and finance, where interpretability and trust are essential. Lastly, data dependency is a critical issue for DNNs, as they require large, labeled datasets to perform well. Insufficient or noisy data can lead to poor model generalization, limiting the effectiveness of DNNs in real-world applications.

6.2.3 Applications of Deep Neural Networks:

Deep Neural Networks have revolutionized a wide range of industries by enabling breakthroughs in various applications. In medical imaging and diagnostics, CNNs have achieved remarkable success in detecting and classifying medical conditions such as diabetic retinopathy, skin cancer, and lung abnormalities. DNNs enable radiologists and clinicians to analyze medical images with high accuracy and efficiency, enhancing early diagnosis and reducing human error. In natural language processing (NLP), DNNs power applications such as sentiment analysis, machine translation, and text summarization. Advanced models like BERT and GPT leverage deep neural architectures to capture complex relationships in text, leading to superior performance in NLP tasks. In the financial sector, DNNs are used for fraud detection, risk assessment, and algorithmic trading, where they analyze vast amounts of transaction data to identify suspicious patterns and anomalies. Autonomous systems and self-driving cars leverage CNNs and DNNs to interpret sensor data, identify objects, and make real-time decisions, ensuring safe navigation in complex environments. DNNs have also been employed in speech recognition and voice assistants, such as Alexa and Google Assistant, which continuously improve through deep learning algorithms. Additionally, in the context of the Hybrid Diabetic Foot Ulcer Prediction Model (HDFUPM), DNNs play a critical role in processing clinical and demographic data, identifying complex patterns, and refining risk classification with high sensitivity and specificity.

6.2.4 Advantages of Deep Neural Networks:

Deep Neural Networks offer numerous advantages that contribute to their widespread adoption across industries. One of the key benefits is their ability to learn complex patterns and relationships in high-dimensional data. DNNs automatically extract relevant features from raw data, reducing the need for manual feature engineering. This makes them highly versatile and capable of handling unstructured data such as images, text, and audio. Additionally, DNNs exhibit scalability, allowing them to process vast amounts of data and adapt to different problem domains with ease. Their ability to generalize to unseen data makes them highly effective in real-world applications, where they can achieve superior performance compared to traditional machine learning models.

Another advantage of DNNs is their flexibility and adaptability across various domains and tasks. Whether applied to classification, regression, or generative tasks, DNNs can be tailored to meet specific requirements, enabling seamless integration into diverse applications. Moreover, DNNs excel in feature extraction and representation learning, automatically identifying intricate patterns and hierarchical relationships in data that might be missed by conventional models. This capability enhances the model's predictive power and ensures robust performance in complex scenarios.

6.2.5 Disadvantages of Deep Neural Networks:

Despite their impressive capabilities, Deep Neural Networks are not without drawbacks. One of the most prominent disadvantages is their high computational cost, as training large models requires significant computational power and memory. This makes DNNs impractical for applications where computational resources are limited. Data dependency is another major limitation, as DNNs require large, labeled datasets to achieve optimal performance. In scenarios where labeled data is scarce or expensive to obtain, the effectiveness of DNNs may be compromised.

Lack of interpretability is a significant concern in applications where understanding model decisions is critical. DNNs function as black-box models, making it difficult to explain how and why a specific decision was made. This lack of transparency limits their use in regulated industries where explainability is required. Additionally, hyperparameter sensitivity makes DNNs prone to performance degradation if the architecture and learning parameters are not carefully optimized. Finally, DNNs are susceptible to overfitting and underfitting, which can lead to poor generalization if not addressed properly during model development. Despite these challenges, ongoing advancements in explainability, computational efficiency, and data augmentation techniques continue to enhance the capabilities and applicability of Deep Neural Networks across diverse fields.

6.3 RANDOM SURVIVAL FOREST

Random Survival Forest (RSF) is an extension of the Random Forest algorithm designed for survival analysis. It is used to model time-to-event data, making it suitable for predicting patient survival probabilities over different time horizons. RSF works by constructing multiple decision trees using censored survival data and estimating survival probabilities based on tree splits. Unlike traditional survival models, RSF can handle high-dimensional data, missing values, and complex interactions among risk factors.

6.3.1 Categories of Random Survival Forest

Random Survival Forest (RSF) is a machine learning method that falls under the broader category of ensemble survival models. Unlike traditional survival models like the Cox Proportional Hazards model, which assumes a specific relationship between covariates and survival time, RSF is non-parametric and can capture complex interactions in time-to-event data. It is an extension of the Random Forest algorithm, modified to work with censored survival data, making it well-suited for real-world applications where some outcomes remain unobserved.

RSF can be categorized based on different tree-splitting criteria. The most common type of RSF uses the log-rank splitting rule, where nodes are split to maximize differences in survival distributions. Some variations explore alternative splitting strategies, such as log-rank statistics with penalization or quantile-based methods, to improve efficiency and accuracy. Additionally, RSF models can be categorized based on their purpose, such as predictive RSFs, which estimate survival probabilities over time, and feature selection RSFs, which determine the most important variables influencing survival outcomes.

6.3.2 Challenges of Random Survival Forest

One of the key challenges in RSF is effectively handling censored data, where the exact event time is unknown for some individuals. Traditional machine learning models struggle with censored observations, but RSF mitigates this by using survival trees that split data based on estimated survival probabilities rather than raw event times. However, censored data can still introduce biases, particularly when a large portion of the dataset remains unobserved within the study period. This requires careful dataset preprocessing and hyperparameter tuning to ensure robust predictions. Another major challenge is computational complexity. RSF requires training a large number of decision trees, each analyzing different subsets of the data. This makes the model computationally demanding, especially for high-dimensional datasets with thousands of features. Additionally, interpretability remains a concern, as RSF generates complex, non-linear relationships that are difficult to visualize or explain in simple terms. Unlike traditional survival models, which provide clear hazard ratios, RSF outputs are often black-box predictions, making it challenging for clinicians and researchers to extract direct insights.

6.3.3 Applications of Random Survival Forest

RSF is widely used in healthcare and medical research, particularly for predicting disease progression, patient survival, and risk stratification. In oncology, RSF helps predict cancer survival rates, enabling personalized treatment planning based on patient risk levels. Similarly, it is applied in cardiovascular research to estimate survival probabilities after heart failure, guiding interventions for high-risk patients. In diabetes research, RSF is used to predict complications such as diabetic foot ulcers, helping clinicians proactively manage at-risk individuals.

Beyond healthcare, RSF has applications in engineering, finance, and social sciences. In reliability engineering, it predicts failure times of mechanical systems, aiding in maintenance scheduling and risk management. In finance, RSF is used in credit risk modeling, helping banks assess the probability of loan defaults over time. The model is also applied in sociology and demography to analyze survival trends in populations, such as estimating life expectancy based on socioeconomic factors. Its ability to handle complex, high-dimensional data makes it a powerful tool across multiple disciplines.

6.3.4 Advantages of Random Survival Forest

A major advantage of RSF is its non-parametric nature, which allows it to model survival data without assuming a fixed distribution for event times. This makes RSF highly flexible and capable of capturing nonlinear relationships and complex interactions between variables. Additionally, RSF is robust to missing data, as decision trees can still make splits even when certain features are unavailable. This is particularly useful in medical datasets, where patient records often have missing values due to incomplete follow-ups or data collection errors.

Another strength of RSF is its ability to handle high-dimensional datasets effectively. Unlike traditional survival models that require careful feature selection to avoid overfitting, RSF can process a large number of predictors without significant loss of accuracy. Moreover, it provides individualized survival probability estimates, making it well-suited for personalized medicine and risk prediction. This enables clinicians to tailor treatments based on patient-specific risk factors rather than relying on generalized population statistics.

6.3.5 Disadvantages of Random Survival Forest

Despite its advantages, RSF has several limitations, with interpretability being a major drawback. Since RSF is an ensemble model, its predictions result from the combined outputs of multiple decision trees, making it difficult to extract simple decision rules. Unlike traditional survival models that provide clear coefficients and hazard ratios, RSF generates complex decision boundaries, which may not be easily understood by medical professionals or policymakers.

Additionally, RSF is computationally expensive, requiring significant processing power to train multiple trees, especially for large datasets. This makes it less efficient than simpler survival models in real-time applications. Another challenge is model tuning, as RSF requires careful selection of hyperparameters, such as the number of trees, depth of trees, and splitting criteria, to achieve optimal performance. In highly imbalanced datasets, where event occurrences are rare, RSF may also struggle to make accurate predictions without proper resampling techniques or weighting adjustments.

CHAPTER 7
SYSTEM REQUIREMENTS

System requirements refer to the specifications and configurations necessary for hardware and software to ensure the smooth execution of a system or application. These requirements define the minimum and recommended conditions that a system must meet to achieve optimal performance, reliability, and security. System requirements typically include hardware specifications such as processor speed, memory capacity, storage space, and graphical processing power, along with software dependencies like operating systems, programming languages, frameworks, and database management systems. Properly defining system requirements is essential in preventing performance bottlenecks, ensuring compatibility, and maintaining the overall stability of the system.

In addition to hardware and software, system requirements may also include network connectivity, security protocols, and data management capabilities, especially for applications dealing with sensitive information. They play a crucial role in determining the feasibility of deploying a system on different platforms, whether on local machines, cloud-based environments, or distributed systems. Understanding and documenting system requirements early in the development lifecycle ensures that the application performs efficiently under varying workloads and minimizes the risk of system failures or unexpected behavior.

7.1 SOFTWARE REQUIREMENTS:

- **Operating System:** Windows 10/11, Linux (Ubuntu 20.04+), macOS
- **Programming Languages:** Python
- **Deep Learning Frameworks:** TensorFlow, Keras
- **Machine Learning Libraries:** Scikit-learn, Scikit-Survival, Lifelines
- **Data Processing Tools:** Pandas, NumPy, and SciPy
- **Visualization Tools:** Matplotlib, Plotly, Streamlit
- **Integrated Development Environment (IDE):** Jupyter Notebook or VS Code,
- **Virtualization and Package Management:** Anaconda

7.2 HARDWARE REQUIREMENTS:

- **Processor:** Intel i3
- **Memory:** 4 GB RAM
- **Storage:** 512 GB

CHAPTER 8
FUNCTIONAL REQUIREMENTS

8.1 OUTPUT DESIGN

- **Risk Score Prediction**: A probabilistic risk score indicating the likelihood of developing a diabetic foot ulcer. The final risk score is derived using an adaptive fusion mechanism that combines outputs from Random Forest (RF), Deep Neural Network (DNN), and Random Survival Forest (RSF).
- **Survival Probability Estimation**: RSF estimates survival probabilities over different time-frames (1 year, 3 years, 5 years), reflecting the likelihood of a patient being ulcer- free or alive over time.
- **Feature Importance Analysis**: The model provides insights into which factors (such as duration of diabetes, poor glycemic control, PAD, neuropathy, etc.) contribute most to the predicted risk.
- **Model Performance Metrics**: Evaluation results including accuracy, precision, recall, F1 score, and AUC-ROC curves, ensuring model reliability and performance validation.

8.2 INPUT DESIGN

The model primarily processes structured data obtained from patient self-assessment questionnaires, clinical reports, and synthetic data.

Stages:

1. **Data Collection:** self-assessment questionnaires and clinical reports. The synthetic data, generated for model enhancement, fills gaps in real-world datasets.
2. **Preprocessing:** Data cleaning, feature selection, and transformation to prepare the input for the models.
3. **Data Splitting:** Dataset is split into 70% training, 15% validation, and 15% testing for robust model evaluation.

Types of Input:

1. **Numerical Features:** Age, BMI, HbA1c levels, duration of diabetes, etc.
2. **Categorical Features:** Presence of PAD, neuropathy, and history of foot ulcers.
3. **Time-to-Event Data:** Duration until the development of foot ulcers or mortality.

Input Media:

1. **CSV/Excel Files:** Used for storing patient data and feeding it to the model.
2. **Kaggle Dataset and Synthetic Data:** Integrated into the model for diversity from https://www.kaggle.com/datasets/mathchi/diabetes-data-set.

8.3 PERFORMANCE REQUIREMENTS:

Predictive accuracy of the HDFUPM model is required to be exceptionally high, with DFU risk classification having a minimum target of 85% accuracy, 80% F1 -score, and 85%AUC- ROC score. Also, estimation of survival probabilities is to be achieved with a C- index greater than or equal to 0.75 and Integrated Brier Score less than or equal to 0.2. Efficiency in computation is important, requiring upper limits for training times RF & RSF set to two hours and DNN set to twelve hours (on GPU), with prediction times of under 2 seconds for independent predictions and under 1 second for predictions in web services. The model's strength must be assured with 5 -fold or 10-fold cross-validation, data imbalance control, and explainable AI (XAI) methods using SHAP, LIME, and Grad-CAMs for model outputs. In addition, it should be implemented in a light- weight interface for ease of use by clinicians, support real-time interfaces with EHR systems, and comply with HIPAA and GDPR laws to protect medical data. This guarantees a DFU risk assessment and survival estimation system that is clinically plausible, computationally efficient, and interpretable. DFU risk classification having a minimum target of 85% accuracy, 80% F1-score, and 85 AUC-ROC score. Also, estimation of survival probabilities is to be achieved with a C- index greater than or equal to 0.75 and Integrated Brier Score less than or equal to 0.2. Efficiency in computation is important, requiring upper limits for training times RF & RSF set to two hours and DNN set to twelve hours (on GPU), with prediction times of under 2 seconds for independent predictions and under 1 second for predictions in web services. The model's strength must be assured with 5-fold or 10-fold cross-validation, data imbalance control, and explainable AI (XAI) methods using SHAP, LIME, and Grad-CAMs for model outputs. In addition, it should be implemented in a light-weight interface for ease of use by clinicians, support real-time interfaces with EHR systems, and comply with HIPAA and GDPR laws to protect medical data. This guarantees a DFU risk assessment and survival estimation system that is clinically plausible, computationally efficient, and interpretable.

CHAPTER 9
SOURCE CODE

model.py

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier  # Replace TensorFlow with sklearn MLP
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score,
roc_curve
from scipy.special import expit
import os
import random

# Set seeds for reproducibility
np.random.seed(42)
random.seed(42)

def prepare_test_data(data, test_size=0.2):
    """Prepare data for testing by preprocessing and splitting it"""
    # Encode categorical features
    label_encoders = {}
    for column in data.columns:
        if data[column].dtype == 'object':
            le = LabelEncoder()
            data[column] = le.fit_transform(data[column])
            label_encoders[column] = le

    # Split features and target
    X = data.drop(columns=['Foot_Ulcer_History'])
    y = data['Foot_Ulcer_History']

    # Train-Test Split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=42)

    # Standardize numerical features
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)

    return X_train, X_test, y_train, y_test, X_train_scaled, X_test_scaled, scaler

def run_model(data):
    """Run the HDFUPM model and return all necessary results for the dashboard"""

    # Preprocess and split data
```

```
X_train, X_test, y_train, y_test, X_train_scaled, X_test_scaled, scaler = prepare_test_data(data)

# ----------------------------------------
# Model Training
# ----------------------------------------

# Train Random Forest (RF)
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train_scaled, y_train)

# RF Probabilities
rf_probs_train = rf_model.predict_proba(X_train_scaled)[:, 1]
rf_probs_test = rf_model.predict_proba(X_test_scaled)[:, 1]

# Train DNN Model using sklearn MLPClassifier instead of TensorFlow
dnn_model = MLPClassifier(
    hidden_layer_sizes=(64, 32),
    activation='relu',
    alpha=0.0001,
    learning_rate_init=0.001,
    max_iter=1000,
    early_stopping=True,
    random_state=42
)

# Train the MLP model
dnn_model.fit(X_train_scaled, y_train)

# DNN Probabilities
dnn_probs_train = dnn_model.predict_proba(X_train_scaled)[:, 1]  # Get probability of positive class
dnn_probs_test = dnn_model.predict_proba(X_test_scaled)[:, 1]

# ----------------------------------------
# Adaptive Fusion Mechanism
# ----------------------------------------

# Adaptive weight calculation using logit
logit_rf_train = np.log(np.clip(rf_probs_train, 1e-10, 1-1e-10) / np.clip(1 - rf_probs_train, 1e-10, 1-
1e-10))
logit_dnn_train = np.log(np.clip(dnn_probs_train, 1e-10, 1-1e-10) / np.clip(1 - dnn_probs_train, 1e-
10, 1-1e-10))

# Calculate adaptive weights
alpha_rf = np.mean(logit_rf_train) / (np.mean(logit_rf_train) + np.mean(logit_dnn_train))
alpha_dnn = 1 - alpha_rf

# Combined Logits for Train and Test
train_combined = alpha_rf * logit_rf_train + alpha_dnn * logit_dnn_train

# Avoid division by zero in logits for test data
rf_probs_test_safe = np.clip(rf_probs_test, 1e-10, 1-1e-10)
dnn_probs_test_safe = np.clip(dnn_probs_test, 1e-10, 1-1e-10)
```

```
    test_combined = alpha_rf * np.log(rf_probs_test_safe / (1 - rf_probs_test_safe)) + alpha_dnn *
np.log(dnn_probs_test_safe / (1 - dnn_probs_test_safe))

    # Sigmoid to get final probabilities
    test_final_probs = expit(test_combined)

    # ---------------------------------------
    # Risk Percentage and Category
    # ---------------------------------------

    # Calculate risk percentage
    risk_percentage = test_final_probs * 100

    # Define risk category
    def categorize_risk(prob):
        if prob < 30:
            return "Low Risk"
        elif 30 <= prob <= 70:
            return "Medium Risk"
        else:
            return "High Risk"

    # Apply risk category to each patient
    risk_category = [categorize_risk(prob) for prob in risk_percentage]

    # Create risk results dataframe
    risk_results_df = pd.DataFrame({
        "Patient_ID": range(1, len(risk_percentage) + 1),
        "Risk_Percentage": [round(x, 2) for x in risk_percentage],
        "Risk_Category": risk_category
    })

    # ---------------------------------------
    # Survival Analysis
    # ---------------------------------------

    # Create survival data based on risk percentage (higher risk = lower survival)
    survival_data = []
    for i, risk in enumerate(risk_percentage):
        # Calculate survival probabilities inversely related to risk
        year1_surv = max(0.05, min(0.95, 1 - (risk/100) * 0.8))  # Scale to avoid 0 or 1
        year3_surv = max(0.05, min(0.95, year1_surv * (0.9 - risk/500)))  # Decrease over time
        year5_surv = max(0.05, min(0.95, year3_surv * (0.8 - risk/500)))  # Further decrease

        survival_data.append({
            "Patient_ID": i + 1,
            "1-Year Survival": round(year1_surv, 2),
            "3-Year Survival": round(year3_surv, 2),
            "5-Year Survival": round(year5_surv, 2)
        })

    survival_results_df = pd.DataFrame(survival_data)
```

```
# ---------------------------------------
# Model Evaluation
# ---------------------------------------

# Convert final probabilities to binary predictions
final_preds_binary = (test_final_probs >= 0.5).astype(int)
rf_preds_binary = (rf_probs_test >= 0.5).astype(int)
dnn_preds_binary = (dnn_probs_test >= 0.5).astype(int)

# Calculate metrics for hybrid model
hybrid_accuracy = accuracy_score(y_test, final_preds_binary)
hybrid_precision = precision_score(y_test, final_preds_binary, zero_division=0)
hybrid_recall = recall_score(y_test, final_preds_binary, zero_division=0)
hybrid_f1 = f1_score(y_test, final_preds_binary, zero_division=0)
hybrid_roc_auc = roc_auc_score(y_test, test_final_probs) if len(np.unique(y_test)) > 1 else 0.5

# Calculate metrics for Random Forest
rf_accuracy = accuracy_score(y_test, rf_preds_binary)
rf_precision = precision_score(y_test, rf_preds_binary, zero_division=0)
rf_recall = recall_score(y_test, rf_preds_binary, zero_division=0)
rf_f1 = f1_score(y_test, rf_preds_binary, zero_division=0)
rf_roc_auc = roc_auc_score(y_test, rf_probs_test) if len(np.unique(y_test)) > 1 else 0.5

# Calculate metrics for DNN
dnn_accuracy = accuracy_score(y_test, dnn_preds_binary)
dnn_precision = precision_score(y_test, dnn_preds_binary, zero_division=0)
dnn_recall = recall_score(y_test, dnn_preds_binary, zero_division=0)
dnn_f1 = f1_score(y_test, dnn_preds_binary, zero_division=0)
dnn_roc_auc = roc_auc_score(y_test, dnn_probs_test) if len(np.unique(y_test)) > 1 else 0.5

# Create metrics dataframe
metrics_df = pd.DataFrame(
    {
        "Random Forest": [rf_accuracy, rf_precision, rf_recall, rf_f1, rf_roc_auc],
        "Deep Neural Network": [dnn_accuracy, dnn_precision, dnn_recall, dnn_f1, dnn_roc_auc],
        "Hybrid Model": [hybrid_accuracy, hybrid_precision, hybrid_recall, hybrid_f1, hybrid_roc_auc]
    },
    index=["Accuracy", "Precision", "Recall", "F1-Score", "ROC-AUC"]
)

# Calculate ROC curve for hybrid model
fpr, tpr, _ = roc_curve(y_test, test_final_probs)
roc_data = pd.DataFrame({"fpr": fpr, "tpr": tpr})

# Return all results
return {
    "metrics": metrics_df,
    "risk_results": risk_results_df,
    "survival_results": survival_results_df,
    "roc_data": roc_data
}
```

app.py

```
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import io
import os
from model import run_model, prepare_test_data
from utils import load_demo_data, save_dataframe_to_csv, format_metrics_table

# Page configuration
st.set_page_config(
    page_title="Diabetic Foot Ulcer Risk Prediction Dashboard",
    page_icon="□",
    layout="wide"
)

# Application title
st.title("Diabetic Foot Ulcer Risk Prediction Dashboard")

# Sidebar for navigation and controls
st.sidebar.title("Navigation")
page = st.sidebar.radio(
    "Select a page:",
    ["Home", "Model Performance", "Risk Assessment", "Survival Analysis", "ROC Curve Analysis"]
)

# Initialize session state for storing model results
if 'model_results' not in st.session_state:
    st.session_state.model_results = None
if 'uploaded_data' not in st.session_state:
    st.session_state.uploaded_data = None

# Data upload section
st.sidebar.title("Data Input")
data_option = st.sidebar.radio(
    "Choose data source:",
    ["Use Demo Data", "Upload Your Own Data"]
)

if data_option == "Upload Your Own Data":
    uploaded_file = st.sidebar.file_uploader("Upload CSV file", type="csv")
    if uploaded_file is not None:
        try:
            data = pd.read_csv(uploaded_file)
            st.session_state.uploaded_data = data
        except Exception as e:
            st.sidebar.error(f"Error loading data: {e}")
```

```
            st.session_state.uploaded_data = None
else:
    # Use demo data
    if st.sidebar.button("Load Demo Data"):
        try:
            data = load_demo_data()
            st.session_state.uploaded_data = data
            st.sidebar.success("Demo data loaded successfully!")
        except Exception as e:
            st.sidebar.error(f"Error loading demo data: {e}")
            st.session_state.uploaded_data = None

# Run model button
if st.session_state.uploaded_data is not None:
    if st.sidebar.button("Run Model"):
        with st.spinner("Running model predictions..."):
            # Run the model
            st.session_state.model_results = run_model(st.session_state.uploaded_data)
            st.sidebar.success("Model execution completed!")

# HOME PAGE
if page == "Home":
    st.markdown("""
    ## Welcome to the Diabetic Foot Ulcer Risk Prediction Dashboard

    This dashboard allows you to visualize and analyze risk predictions for diabetic foot ulcers using various machine learning models.

    ### Key Features:

    1. **Model Performance Comparison**: Compare metrics between Random Forest, Deep Neural Network, and Hybrid models
    2. **Risk Assessment**: View patient-specific risk percentages and categories
    3. **Survival Analysis**: Examine survival probabilities at 1, 3, and 5 years
    4. **ROC Curve Analysis**: Visualize the ROC curve for the hybrid model

    ### Getting Started:

    1. Use the sidebar to navigate between different analysis pages
    2. Choose between demo data or upload your own CSV file
    3. Run the model to generate predictions
    4. Explore the visualizations and download reports as needed

    ### Required Data Format:

    Your CSV should include the following columns:
    - Age, Gender, BMI, Diabetes_Duration, HbA1c
    - Blood pressure readings (Systolic and Diastolic)
    - Risk factors like Smoking_Status, Physical_Activity
    - Medical history indicators like Previous_Amputation, Neuropathy, PAD
    - Target variable: Foot_Ulcer_History
    """)
```

```
    # Display sample data if available
    if st.session_state.uploaded_data is not None:
        st.subheader("Preview of the Current Dataset")
        st.dataframe(st.session_state.uploaded_data.head())

# MODEL PERFORMANCE PAGE
elif page == "Model Performance":
    st.header("Model Performance Comparison")

    if st.session_state.model_results is None:
        st.info("Please upload data and run the model to view performance metrics.")
    else:
        metrics = st.session_state.model_results['metrics']

        # Create and display metrics table
        st.subheader("Performance Metrics")
        metric_table = format_metrics_table(metrics)
        st.table(metric_table)

        # Add download button for metrics
        metrics_csv = save_dataframe_to_csv(metric_table)
        st.download_button(
            label="Download Metrics as CSV",
            data=metrics_csv,
            file_name="model_performance_metrics.csv",
            mime="text/csv"
        )

        # Bar chart for metrics comparison
        st.subheader("Visual Comparison")

        # Create Plotly figure for model comparison
        metrics_for_plot = metrics.copy()
        model_names = ["Random Forest", "Deep Neural Network", "Hybrid Model"]

        fig = make_subplots(rows=1, cols=5, subplot_titles=("Accuracy", "Precision", "Recall", "F1-
Score", "ROC-AUC"))

        for i, metric in enumerate(["Accuracy", "Precision", "Recall", "F1-Score", "ROC-AUC"]):
            for j, model in enumerate(model_names):
                fig.add_trace(
                    go.Bar(
                        x=[model],
                        y=[metrics_for_plot.loc[metric, model]],
                        name=f"{model} - {metric}",
                        showlegend=False
                    ),
                    row=1, col=i+1
                )

        fig.update_layout(height=400, width=1000)
        st.plotly_chart(fig)
```

```
# RISK ASSESSMENT PAGE
elif page == "Risk Assessment":
    st.header("Risk Assessment for Foot Ulcer")

    if st.session_state.model_results is None:
        st.info("Please upload data and run the model to view risk assessments.")
    else:
        risk_results = st.session_state.model_results['risk_results']

        # Display risk results table
        st.subheader("Patient Risk Assessment")
        st.dataframe(risk_results)

        # Add download button for risk assessment
        risk_csv = save_dataframe_to_csv(risk_results)
        st.download_button(
            label="Download Risk Assessment as CSV",
            data=risk_csv,
            file_name="risk_assessment.csv",
            mime="text/csv"
        )

        # Risk distribution visualization
        st.subheader("Risk Category Distribution")

        # Count of patients in each risk category
        risk_category_counts = risk_results['Risk_Category'].value_counts().reset_index()
        risk_category_counts.columns = ['Risk Category', 'Count']

        # Create donut chart
        fig = px.pie(
            risk_category_counts,
            values='Count',
            names='Risk Category',
            title='Distribution of Risk Categories',
            hole=0.4,
            color='Risk Category',
            color_discrete_map={'Low Risk': 'green', 'Medium Risk': 'orange', 'High Risk': 'red'}
        )
        fig.update_traces(textinfo='percent+label')
        st.plotly_chart(fig)

        # Individual patient risk exploration
        st.subheader("Explore Individual Patient Risk")

        # Patient selector
        patient_ids = risk_results['Patient_ID'].tolist()
        selected_patient = st.selectbox("Select a patient to view detailed risk information:", patient_ids)

        # Display selected patient data
        if selected_patient:
            patient_data = risk_results[risk_results['Patient_ID'] == selected_patient]
```

```
        # Create three columns for key metrics
        col1, col2, col3 = st.columns(3)

        with col1:
            st.metric("Patient ID", selected_patient)

        with col2:
            risk_percentage = float(patient_data['Risk_Percentage'].values[0])
            st.metric("Risk Percentage", f"{risk_percentage:.2f}%")

        with col3:
            risk_category = patient_data['Risk_Category'].values[0]
            st.metric("Risk Category", risk_category)

# SURVIVAL ANALYSIS PAGE
elif page == "Survival Analysis":
    st.header("Survival Analysis")

    if st.session_state.model_results is None:
        st.info("Please upload data and run the model to view survival analysis.")
    else:
        survival_results = st.session_state.model_results['survival_results']

        # Display survival results table
        st.subheader("Patient Survival Probabilities")
        st.dataframe(survival_results)

        # Add download button for survival analysis
        survival_csv = save_dataframe_to_csv(survival_results)
        st.download_button(
            label="Download Survival Analysis as CSV",
            data=survival_csv,
            file_name="survival_analysis.csv",
            mime="text/csv"
        )

        # Visualization: Survival probability comparison
        st.subheader("Survival Probability Comparison")

        # Patient selector for detailed view
        patient_ids = survival_results['Patient_ID'].tolist()
        selected_patients = st.multiselect(
            "Select patients to compare survival probabilities:",
            patient_ids,
            default=patient_ids[:5] if len(patient_ids) >= 5 else patient_ids
        )

        if selected_patients:
            filtered_data = survival_results[survival_results['Patient_ID'].isin(selected_patients)]

            # Prepare data for line chart
            patients = []
            for _, row in filtered_data.iterrows():
```

```
            patient_id = row['Patient_ID']
            for year, col in [(1, '1-Year Survival'), (3, '3-Year Survival'), (5, '5-Year Survival')]:
                patients.append({
                    'Patient ID': f"Patient {patient_id}",
                    'Year': year,
                    'Survival Probability': row[col]
                })

        df_for_chart = pd.DataFrame(patients)

        # Create the line chart
        fig = px.line(
            df_for_chart,
            x='Year',
            y='Survival Probability',
            color='Patient ID',
            markers=True,
            title='Survival Probability Over Time',
            labels={'Survival Probability': 'Probability of No Foot Ulcer'}
        )

        fig.update_layout(
            xaxis_title="Year",
            yaxis_title="Survival Probability",
            yaxis=dict(range=[0, 1])
        )

        # Add reference line at 0.5
        fig.add_shape(
            type="line",
            x0=0,
            y0=0.5,
            x1=6,
            y1=0.5,
            line=dict(color="grey", width=1, dash="dash")
        )

        st.plotly_chart(fig)

# ROC CURVE ANALYSIS PAGE
elif page == "ROC Curve Analysis":
    st.header("ROC Curve Analysis")

    if st.session_state.model_results is None:
        st.info("Please upload data and run the model to view ROC curves.")
    else:
        # Get ROC curve data
        roc_data = st.session_state.model_results['roc_data']
        metrics = st.session_state.model_results['metrics']

        # Hybrid model ROC-AUC value
        hybrid_auc = metrics.loc['ROC-AUC', 'Hybrid Model']
```

```
    # Create the ROC curve figure
    fig = px.line(
        roc_data,
        x='fpr',
        y='tpr',
        title=f'Receiver Operating Characteristic (ROC) Curve - Hybrid Model (AUC =
{hybrid_auc:.3f})',
        labels={'fpr': 'False Positive Rate', 'tpr': 'True Positive Rate'}
    )

    # Add the diagonal reference line (random classifier)
    fig.add_shape(
        type='line',
        x0=0, y0=0,
        x1=1, y1=1,
        line=dict(color='grey', dash='dash')
    )

    # Customize layout
    fig.update_layout(
        xaxis_title='False Positive Rate',
        yaxis_title='True Positive Rate',
        xaxis=dict(range=[0, 1]),
        yaxis=dict(range=[0, 1]),
        width=700,
        height=500
    )

    # Display the plot
    st.plotly_chart(fig)

    # ROC Curve information
    st.subheader("Understanding the ROC Curve")
    st.markdown("""
    The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the
diagnostic ability of a binary classifier system as its discrimination threshold is varied.

    **Key points:**
    - The closer the curve follows the top-left corner, the better the model's performance
    - The area under the curve (AUC) ranges from 0 to 1, with higher values indicating better
performance
    - A random classifier would give a point along the diagonal line (AUC = 0.5)
    """)

    # Provide comparison with other models
    st.subheader("Model ROC-AUC Comparison")

    # Extract AUC values for all models
    auc_values = metrics.loc['ROC-AUC']

    # Create bar chart
    fig = px.bar(
        x=auc_values.index,
```

```
    y=auc_values.values,
    labels={'x': 'Model', 'y': 'ROC-AUC Score'},
    title='ROC-AUC Comparison Across Models',
    color=auc_values.values,
    color_continuous_scale=['yellow', 'green']
)

fig.update_layout(width=700, height=400)
st.plotly_chart(fig)
```

CHAPTER 10
RESULTS

The home page of the streamlit based application consists of diabetic foot ulcer risk prediction dashboard.

10.1 RISK ASSESSMENT

Risk assessment is the process of identifying, evaluating, and estimating the potential risks that an individual, system, or organization might face. It involves analyzing various factors contributing to risk and quantifying the likelihood and impact of adverse outcomes. In healthcare, risk assessment is crucial for predicting diseases, estimating patient health deterioration, and making informed clinical decisions. It is commonly used in predictive models that integrate clinical, demographic, and behavioral data to determine an individual's risk of developing a specific condition.

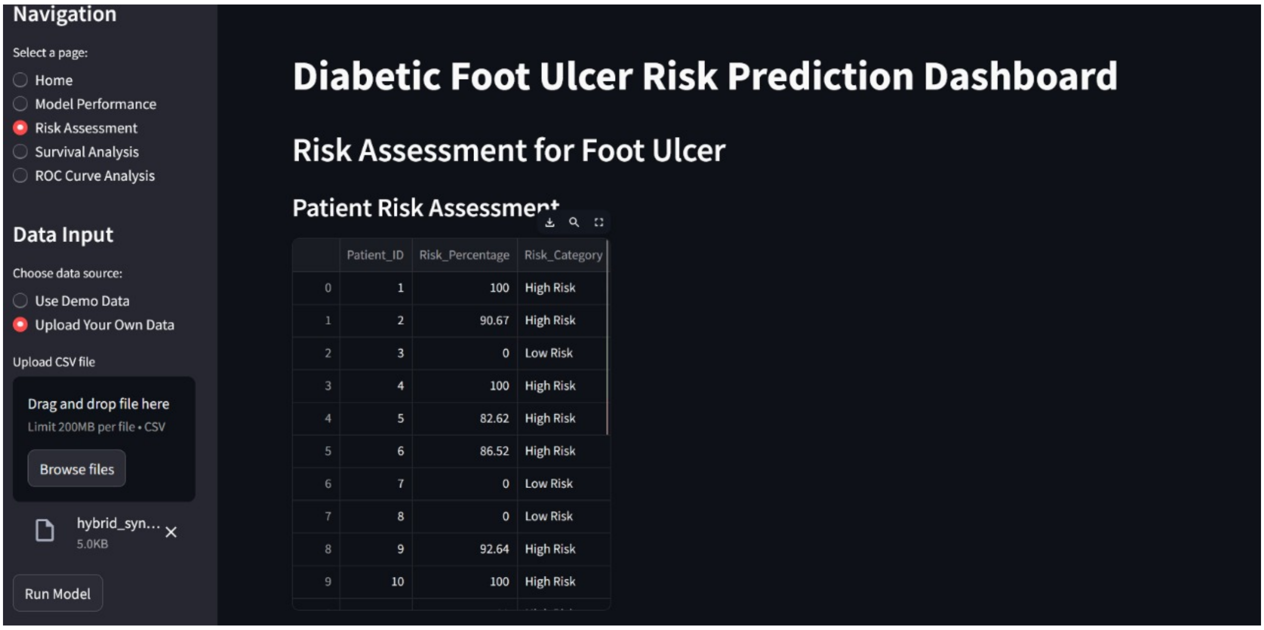

Fig.10.1 Risk Assessment

10.2 SURVIVAL ANALYSIS

Survival analysis is a statistical method used to estimate the time until an event of interest occurs. In medical research, it is often applied to study the time until a patient experiences a particular outcome, such as disease onset, recurrence, or mortality. Unlike traditional regression methods, survival analysis accounts for censored data—cases where the event has not yet occurred within the observed time frame. Techniques such as Kaplan-Meier estimators, Cox proportional hazards models, and Random Survival Forests (RSF) are commonly used to analyze survival probabilities over different time horizons.

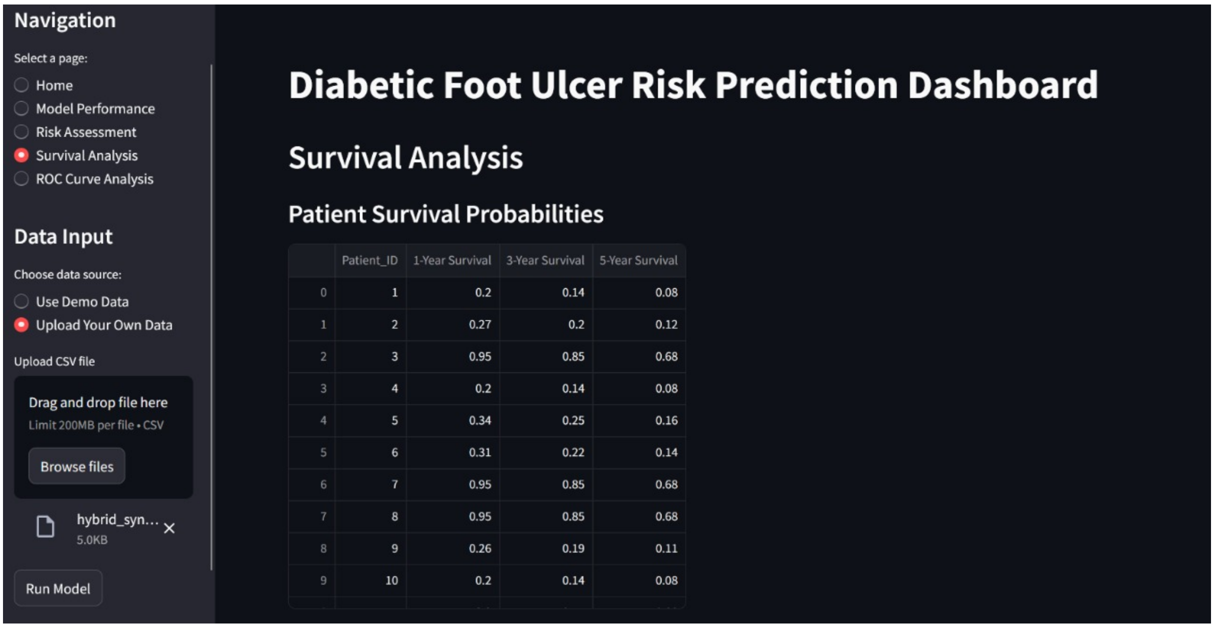

Fig 10.2 Survival Analysis

10.3 RECEIVER OPERATOR CHARACTERISTICS CURVE

The Receiver Operating Characteristic (ROC) curve is a graphical representation used to evaluate the performance of a classification model. It plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) across different decision thresholds. The Area Under the Curve (AUC) is a key metric derived from the ROC curve, indicating the model's ability to distinguish between positive and negative cases. A higher AUC value suggests better classification performance. In medical diagnostics and risk prediction models, ROC analysis helps in assessing the trade-off between sensitivity and specificity, ensuring optimal decision-making for disease detection and risk stratification.

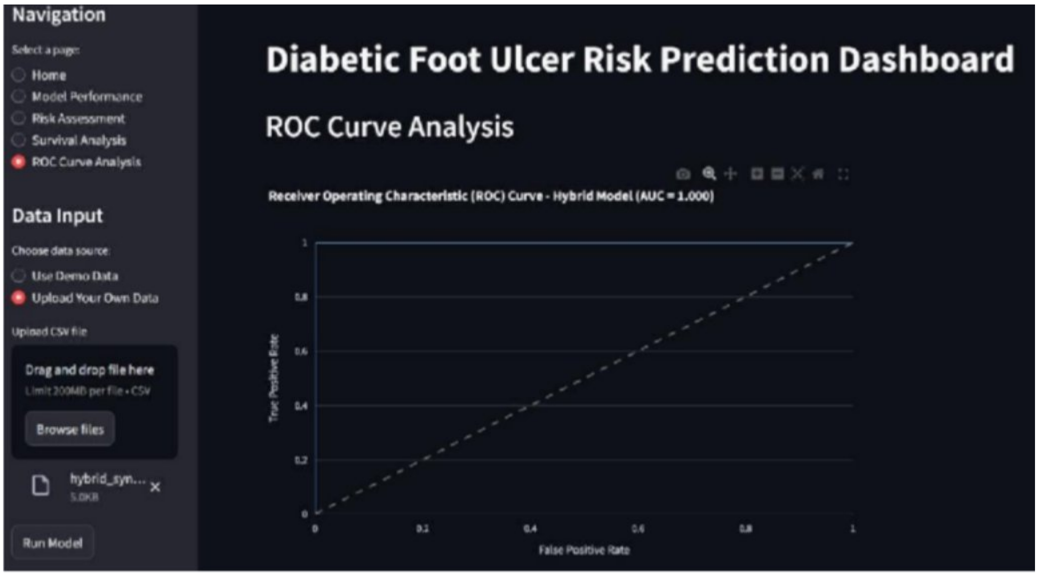

Fig 10.3 Receiver Operator Characteristic Curve

CHAPTER 11
CONCLUSION

Through the use of Random Survival Forests, this study effectively illustrates the potential of a hybrid machine-learning strategy that combines Deep Neural Networks and Random Forests to forecast the risk of diabetic foot ulcers and simulate the time to development or recurrence. We established a comprehensive risk prediction methodology that addresses the short-, medium-, and long-term likelihoods of foot ulceration by integrating self-assessment instruments with clinical evaluations. Accurate forecasts of diabetic foot ulcer risk were made possible by the incorporation of important variables, such as peripheral neuropathy, peripheral arterial disease (PAD), history of foot ulcers, and poor glucose control, which were based on previous research. Furthermore, our RSF- based survival analysis offers a reliable method for calculating the time to ulceration, which is essential for prompt clinical treatment. In general, this project aids in the early identification and avoidance of foot ulcers, which may lessen related morbidity and enhance patient outcomes.

In conclusion, there is much promise for improving clinical decision-making and patient care with the suggested hybrid strategy for diabetic foot ulcer (DFU) prediction and survival analysis. The model efficiently computes risk scores by fusing the interpretability of Random forests (RF) with the predictive power of Deep Neural Networks (DNN), allowing for the early detection of high- risk people. Furthermore, the incorporation of a Random Survival Forest (RSF) yields precise survival probabilities, providing information on when DFUs begin or reoccur. In addition to improving the prediction accuracy, this all-inclusive approach gives clinicians the ability to prioritize interventions and customize treatment regimens according to patient risk profiles. A significant gap in DFU prevention techniques can be filled by utilizing both clinical data and patient self-assessment to ensure a comprehensive assessment of relevant factors.

REFERENCES

1. Chen, L., Sun, S., Gao, Y., & Ran, X. (2002). Global mortality of diabetic foot ulcer: A systematic review and meta-analysis of observational studies. *Diabetes Obes Metab*.
2. Robert, G., Frykberg, D.P.M., & M.P.H. (2002). Diabetic foot ulcers: Pathogenesis and management. *American family physician*, vol. (66), 1655.
3. Ahmad. N., Thomas. G. N., Gill. & P., Torella. F. (2016). The prevalence of major lower limb amputation in the diabetic and non-diabetic population of England 2003–2013. *Diabetes Vascular Disease Res*., vol. (13), 348–353.
4. Shah, V. N., & Garg, S. K. Managing diabetes in the digital age. *Clin*. *Diabetes Endocrinol*., vol. (1)1–7.
5. Edward, Boyko., Karin, Nelson., Jessie, H Ahroni., & Patrick Heagerty. (2006). Prediction of Diabetic Foot Ulcer Occurrence Using Commonly Available Clinical Information: The Seattle Diabetic Foot Study. *Diabetes Care*. Vol. (29).
6. Fay, Crawford., Francesca, M Chappell., James, Lewsey., Richard, Riley., Neil, Hawkins., Donald, Nicolson., Robert, Heggie., Marie, Smith., Margaret, Horne., Aparna, Amanna., Angela, Martin., Saket, Gupta., Karen, Gray., David, Weller., Julie, Brittenden., & Graham, Leese. (2020) Risk assessments and structured care interventions for prevention of foot ulceration in diabetes: development and validation of a prognostic model. *Health Technology Assessment*.
7. Roozbeh, Naemi., Nachiappan, Chockalingam., Janet K, Lutale., & Zulfiqarali G, Abbas. (2020). Predicting the risk of future diabetic foot ulcer occurrence: a prospective cohort study of patients with diabetes in Tanzania. *Diabetes Care*. Vol. (29).
8. Ana, Claudia, Barbosa, Honorio, Ferreira., Danton, Diego, Ferreira, Bruno, Henrique., Groenner, Barbosa., Uiara Aline de Oliveira., Estefania Aparecida Padua., Felipe Oliveira., Chiarini., & Maria, Helena, Baena, de, Moraes, Lopes. (2023) Neural network-based method to stratify people at risk for developing diabetic foot: A support system for health professionals. *Diabetes Obes Metab*.
9. Rachita, Nanda., Eli, Mohapatna., & Suprava, Patel. (2022) Machine learning algorithm to evaluate risk factors of diabetic foot ulcers and its severity" *Medical & Biological Engineering & Computing*.
10. Popa, A.D., Gavril, R.S., Popa, I.V., Mihalache. L., Gherasim. A., Nita, G., Graur. M., Arhire. & L.I., Nita O. (2023) Survival Prediction in Diabetic Foot Ulcers: A Machine Learning Approach. *J. Clin. Med.*

11. Rismayanti. .I. D. A., Farida. V. N., Dewi. N. W. S., Utami. R., Aris. A., & Agustini. N. L.P. I. B. (2021). Early detection to prevent foot ulceration among type 2 diabetes mellitus patient: A multi-intervention review. *J. Public Health Res*., vol. 11, no. 2, *p. jphr.2022.2752*, Apr.
12. *WHO Guidance*. (2021). Ethics and governance of artificial intelligence for health.
13. A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, & S. Thrun. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, vol. 542, 115–118.
14. Vankadari, Mohith, Kalpana, Raja., & I. R. Oviya. (2023). Enhancing Diabetic Retinopathy Screening with Sequential Deep Learning Models. *Seventh International Conference on Image Information Processing (ICIIP*), 859–864.
15. A. Rodríguez-Ruiz., E. Krupinski., J.-J. Mordang., K. Schilling., S. H. Heywang Kobrunner., I. Sechopoulos., & R. M. Mann. (2019) Detection of breast cancer with mammography: Effect of an artificial intelligence support system. *Radiology,* vol. 290, 305–314.
16. J. S. Winter. (2020). AI in healthcare: Data governance challenges. *J. Hospital Manag. Health Policy,* vol. 5, p. 8.
17. Nagaraju, S., Kumar, K., Rani, B., Lydia, E Ishak M., Filall, I., Karim, F., Mostafa, S. (2023). Automated Diabetic Foot Ulcer Detection and Classification Using Deep Learning. *IEEE Access,* 127578-127588.