ABSTRACT

Early detection of Chronic Kidney Disease (CKD) remains a vital challenge in healthcare, as the disease often progresses silently until advanced stages. This project presents an AI-powered clinical decision support system aimed at improving diagnostic speed, accuracy, and accessibility. The pipeline begins with preprocessing of structured clinical data, including normalization, missing value imputation, and categorical encoding to ensure data quality and consistency. A focused subset of nine clinically relevant biomarkers were selected to reduce complexity without sacrificing predictive performance. The core model is a Multi-Layer Perceptron (MLP) neural network comprising two hidden layers with ReLU activation, batch normalization, and dropout regularization, optimized using the Adam optimizer and trained on a binary cross-entropy loss function. Model performance is evaluated using standard metrics including confusion matrix, precision, recall, and F1-score. This system demonstrates the viability of deep learning for structured health data and highlights its potential role in early-stage CKD screening. The solution is lightweight, clinically interpretable, and scalable for integration into real-world electronic health systems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

S. NO.	ABBREVATIONS	EXPANSIONS
1	AI	Artificial Intelligence
2	CKD	Chronic Kidney Disease
3	ML	Machine Learning
4	MLP	Multi-Layer Perceptron
5	EMR	Electronic Medical Record
6	EHR	Electronic Health Record
7	DM	Diabetes Mellitus
8	SG	Specific Gravity
9	AL	Albumin
10	HEMO	Hemoglobin
11	PVC	Packed Cell Volume
12	HTN	Hypertension
13	RBC	Red Blood Cell Count
14	ReLU	Rectified Linear Unit
15	BCE	Binary Cross Entropy
16	LIME	Local Interpretable Model-Agnostic Explanations
17	CDSS	Clinical Decision Support System
19	TP	True Positive
20	TN	True Negative
21	FP	False Positive
22	FN	False Negative
23	SC	Serum Creatinine
24	CSV	Comma Separated Values

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

Chronic Kidney Disease (CKD) is a progressive and potentially life-threatening condition that often goes undetected in its early stages due to mild or absent symptoms. Early diagnosis is crucial to prevent irreversible kidney damage and the need for long-term dialysis or transplantation. However, traditional diagnostic methods can be resource-intensive and slow, particularly in underserved regions. In recent years, artificial intelligence (AI) and deep learning have shown significant promise in clinical decision support, enabling faster and more accurate disease detection using routine health data.

This project aims to develop an AI-powered prediction system for the early detection of CKD using structured clinical data. The solution integrates preprocessing techniques to handle missing values and standardize features, followed by a Multi-Layer Perceptron (MLP) model trained on nine key clinical inputs. The model is optimized for high accuracy in binary classification tasks. This approach reduces diagnostic dependency on extensive lab testing and supports early screening efforts. The system is designed to be lightweight, scalable, and suitable for deployment in clinical environments as a reliable diagnostic aid. It ensures consistent and real-time prediction, making it valuable for both routine checkups and proactive screening. The model's ability to generalize across patient profiles enhances its usability, and its integration into a user interface allows for accessible and intuitive interaction for medical professionals.

1.2 OBJECTIVE OF THE PROJECT

- To develop a deep learning-based diagnostic model for structured clinical data
- To preprocess the dataset by handling missing values, encoding categorical features, and normalizing inputs
- To select nine key clinical attributes for efficient and accurate CKD prediction
- To implement a Multi-Layer Perceptron (MLP) model with hidden layers, dropout, and batch normalization
- To train the model using binary cross-entropy loss and the Adam optimizer
- To evaluate the model using metrics such as accuracy, precision, recall, and F1-score
- To achieve high prediction accuracy for classifying CKD-positive and CKD-negative cases
- To create a scalable and lightweight system suitable for integration into clinical workflows

1.3 SCOPE OF THE PROJECT

This project focuses on applying artificial intelligence to enhance early-stage detection of Chronic Kidney Disease through structured clinical data analysis. It demonstrates the integration of deep learning specifically Multi-Layer Perceptron model into a practical medical diagnostic pipeline. The system is optimized for minimal input requirements, high accuracy, and ease of deployment, making it particularly suitable for use in screening environments and primary care settings. Rather than relying on complex imaging or invasive procedures, the model utilizes standard clinical parameters that are commonly collected during routine checkups.

Designed with flexibility and scalability in mind, the framework can be extended to other medical conditions involving tabular patient data, with minimal architectural changes. While this implementation is tailored for CKD, it serves as a broader demonstration of how AI can support early detection and improve clinical decision-making.

CHAPTER 2

LITERATURE SURVEY

2.1. INTRODUCTION

Chronic Kidney Disease (CKD) is a progressive illness that often remains undiagnosed until its advanced stages, making early detection crucial. Traditional diagnostic methods are effective but can be expensive and inaccessible in certain regions. With the rise of machine learning and artificial intelligence, researchers have developed predictive models using routine clinical data to improve early detection. Various algorithms, including decision trees, support vector machines, and neural networks, have been applied with promising results. The following studies highlight key developments in AI-based CKD prediction.

2.2 REVIEW OF EXISTING LITERATURE

Paper 1:

Title: "Clinically Applicable Machine Learning Approaches to Identify Attributes of Chronic Kidney Disease (CKD) for Use in Low-Cost Diagnostic Screening"

Authors: Md. Rashed-Al-Mahfuz, Abedul Haque, AKM Azad, Salem A. Alyami, Julian M. W. Quinn, and Mohammad Ali Moni

Year: 2023

The paper explores the application of machine learning for identifying a reduced set of clinical attributes that can effectively predict CKD in a cost-

efficient manner. It utilizes the UCI CKD dataset alongside hospital-collected data to train multiple classifiers, including Random Forest, XGBoost, Gradient Boosting, Logistic Regression, and SVM. The models are evaluated using various performance metrics, with Random Forest achieving the highest accuracy. To ensure clinical transparency, the study employs SHAP (Shapley Additive explanations) to interpret model predictions and identify the most important features. Through this interpretability framework, the number of required clinical inputs is reduced from 24 to 13 without compromising predictive performance. The approach is practical, interpretable, and geared toward deployment in real-world healthcare settings. However, the paper does not implement a deep learning model or explore real-time system deployment. The present project addresses these gaps by using a reduced set of 8 clinical inputs, a deep learning-based MLP model for prediction, and a deployable architecture aimed at early CKD screening with high accuracy and minimal resource usage.

Paper 2:

Title: "A Reduced Set of Features for Chronic Kidney Disease Prediction"

Authors: Rajesh Misir, Malay Mitra, Ranjit Kumar Samanta

Year: 2022

This paper investigates the potential for predicting Chronic Kidney Disease (CKD) using a minimal number of clinical features, with the goal of reducing diagnostic complexity and cost. The authors employ the UCI CKD dataset and apply Correlation-based Feature Selection (CFS) combined with various search strategies, including Best First and Genetic Search, to identify the most informative attributes. As a result, nine significant features are selected, such as serum creatinine, hemoglobin, specific gravity, and albumin. The selected features are then evaluated using two neural network-based classifiers

Incremental Backpropagation Learning Network (IBPLN) and Levenberg Marquardt (LM) algorithm. Both classifiers demonstrate high classification performance, with LM achieving a slightly higher accuracy and sensitivity. The study emphasizes reducing input features without sacrificing model quality, making it suitable for practical clinical use. However, it does not implement dropout, batch normalization, or a full deep learning pipeline. The current project extends this work by integrating dropout regularization, batch normalization, and a Multi-Layer Perceptron (MLP) architecture, achieving comparable accuracy using the same reduced input strategy, with improved model robustness and training stability.

Paper 3:

Title: "Machine Learning Hybrid Model for the Prediction of Chronic Kidney Disease"

Authors: Hira Khalid, Hiba Khalid, Aimen Basharat, Hafiza Tooba Bukhari

Year: 2023

This paper presents a hybrid machine learning approach for the accurate prediction of Chronic Kidney Disease (CKD) using an ensemble of base classifiers and a meta-classifier. The authors focus on enhancing prediction accuracy through model stacking, where classifiers such as Gaussian Naïve Bayes, Gradient Boosting, and Decision Trees are used as base models, and a Random Forest model serves as the meta-classifier. The UCI CKD dataset is used, and Pearson correlation analysis is applied for initial feature selection to reduce redundancy and improve model training. The authors compare the performance of individual classifiers and the stacked ensemble. While Gradient Boosting and Random Forest individually achieved 99% and 98% accuracy, respectively, the proposed hybrid model outperformed all others with a perfect

100% accuracy on the dataset. However, the paper does not explore deep learning methods, interpretability techniques, or clinical deployment strategies.

Paper 4:

Title: "Chronic Kidney Disease Prediction Based on Machine Learning Algorithms"

Authors: Md. Ariful Islam, Md. Sajidul Islam, Md. Mostafa Kamal, Mohammad Shahin Mia, Prabir Kumar Biswas

Year: 2023

This paper presents a comparative analysis of multiple machine learning algorithms to predict Chronic Kidney Disease (CKD) using the UCI CKD dataset. The authors apply a thorough preprocessing pipeline that includes missing value imputation, label encoding, and dimensionality reduction using Principal Component Analysis (PCA). A total of twelve machine learning algorithms are tested, including XGBoost, CatBoost, LightGBM, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression, and Artificial Neural Networks (ANN). Among all the models, XGBoost achieved the highest accuracy of 98.3%, along with strong F1-score and precision values. The study also emphasizes the importance of minimizing input features while maintaining classification quality, making the approach suitable for real-time clinical applications. Although the results are strong, the paper does not explore advanced neural network architectures or interpretability mechanisms. The present project builds on this work by applying a deep learning-based Multi-Layer Perceptron (MLP) model that achieves over 99% accuracy using only nine selected clinical features, and incorporates training enhancements like dropout and batch normalization for improved model reliability and generalization.

Paper 5:

Title: "Artificial Intelligence Predicts the Progression of Diabetic Kidney Disease Using Big Data Machine Learning"

Authors: Masaki Makino, Ryo Yoshimoto, Masaki Ono, Hideki Itoko, Kenichiro Katsuki, Tomohiro Koseki, Kengo Hirata, Kenta Ueki, Hiroshi Iwata, Satoshi Kaneko, Hiroshi Kajio

Year: 2019

This paper proposes a machine learning framework for predicting the progression of Diabetic Kidney Disease (DKD), leveraging large-scale electronic medical record (EMR) data from over 64,000 patients. The authors use deep learning techniques, including convolutional autoencoders, for feature representation, followed by a logistic regression classifier for prediction. The system processes both structured time-series data and clinical events over a six-month window to generate long-term progression forecasts. The model identifies patients likely to experience DKD progression with approximately 71% accuracy and provides insights into associated risks such as dialysis and cardiovascular complications. The research demonstrates the utility of AI in long-term kidney disease forecasting using real-world clinical data, particularly through its use of unsupervised pre-training and EMR-driven learning. However, the model operates as a temporal forecasting tool rather than a direct diagnostic classifier, and it lacks integration of traditional CKD features such as lab results. In contrast, the present project implements a supervised learning pipeline focused on early CKD detection using a compact clinical feature set and a Multi-Layer Perceptron (MLP), offering an interpretable and deployable solution tailored for early screening and binary classification.

Paper 6:

Title: "A Deep Neural Network for Early Detection and Prediction of Chronic Kidney Disease"

Authors: Vijendra Singh, Vijayan K. Asari, Rajkumar Rajasekaran

Year: 2022

This paper presents a deep learning approach to predict Chronic Kidney Disease (CKD) using a carefully designed Deep Neural Network (DNN) architecture. The authors use the UCI CKD dataset and apply extensive preprocessing steps, including mean imputation for missing values and outlier handling. Feature selection is performed using Recursive Feature Elimination (RFE) to identify the most relevant predictors. The proposed DNN comprises 12 layers, including dense layers, dropout layers for regularization, and ReLU activation functions. It is trained using binary cross-entropy loss and the Adam optimizer. The model outperforms traditional classifiers such as Support Vector Machines (SVM), K-Nearest Neighbor (KNN), and Random Forest, achieving 100% accuracy on the test dataset. The paper demonstrates the power of deep learning in handling structured medical data and highlights the importance of robust preprocessing, architecture design, and model validation. However, it does not include interpretability techniques or a user-facing deployment aspect. The current project extends this foundation by building a more lightweight and interpretable MLP model with dropout and batch normalization, achieving similar performance using fewer layers and a reduced clinical feature set, making it more suitable for real-time screening in clinical environments.

Paper 7:

Title: “Deep Learning-Based Diagnosis of Chronic Kidney Disease Using Clinical Data”

Authors: Fábio V. de S. Lucas, André C. P. L. F. de Carvalho

Year: 2022

This paper investigates the application of deep learning techniques for the diagnosis of Chronic Kidney Disease (CKD) using structured clinical datasets. The authors use the UCI CKD dataset and apply a comprehensive preprocessing workflow, including imputation of missing values, label encoding of categorical attributes, and feature normalization. Several deep learning models are evaluated, including fully connected networks with varying depths and neuron counts. The study also experiments with different combinations of activation functions, optimization strategies, and regularization techniques such as dropout and batch normalization. The best-performing model achieves over 99% accuracy, outperforming classical machine learning algorithms like Decision Trees and Naïve Bayes. Additionally, the paper emphasizes the importance of hyperparameter tuning and architecture selection to optimize deep learning model performance. While the paper successfully demonstrates the high predictive capacity of deep networks for CKD detection, it lacks an emphasis on model explainability or real-time deployment. The present project builds upon this work by implementing a focused Multi-Layer Perceptron (MLP) with optimized architecture, using a reduced set of nine clinical features, and maintaining high accuracy while also aiming for a lightweight and clinically integrable model design.

Paper 8:

Title: "On the Diagnosis of Chronic Kidney Disease Using a Machine Learning-Based Interface with Explainable Artificial Intelligence"

Authors: Gangani Dharmarathne, Madhusha Bogahawaththa, Marion McAfee, Upaka Rathnayake, D.P.P. Meddage

Year: 2024

This paper introduces a comprehensive machine learning approach integrated with explainable artificial intelligence (XAI) to diagnose Chronic Kidney Disease (CKD) and improve trust in clinical decisions. Using the UCI CKD dataset, six ML algorithms Decision Tree, K-Nearest Neighbors, Support Vector Machine, Random Forest, Extreme Gradient Boost (XGB), and Artificial Neural Network were trained and evaluated. Among them, XGB yielded the highest accuracy. The authors emphasized explainability by applying SHAP (Shapley Additive Explanations) and Partial Dependency Plots (PDP) to provide transparency into the prediction rationale. Data preprocessing included KNN based imputation and feature selection based on missing value thresholds. A novel feature of the study was the development of a graphical user interface (GUI) that not only predicts CKD but also visually justifies predictions using XAI, supporting clinicians in diagnosis and decision-making.While the proposed architecture focuses on privacy and explainability at a system level, the present project addresses data efficiency and prediction reliability by implementing a streamlined, centralized MLP model that is practical for direct clinical use and early detection. The paper demonstrates the power of deep learning in handling structured medical data and highlights the importance of robust preprocessing, architecture design, and model validation.

Paper 9:

Title: "Privacy-First ML for Chronic Kidney Disease Prediction: Exploring a Decentralized Approach Using Blockchain and IPFS"

Authors: Vikram Puri, Ishaani Priyadarshini, Aman Kataria, Sita Rani, Hong Min

Year: 2024

This paper proposes a privacy-preserving, decentralized machine learning framework for Chronic Kidney Disease (CKD) prediction, combining blockchain, InterPlanetary File System (IPFS), and Explainable AI (XAI). Addressing the challenge of sensitive healthcare data sharing, the study introduces a decentralized collaborative model where healthcare institutions can train ML models locally and exchange only model updates over the blockchain. The approach uses Extreme Gradient Boosting (XGBoost) for prediction and LIME for explanation. Experiments were conducted on two datasets one public and another from a hospital with results showing high accuracy and interpretability. The architecture ensures patient data remains local, significantly enhancing privacy and robustness. While the proposed architecture focuses on privacy and explainability at a system level, the present project addresses data efficiency and prediction reliability by implementing a streamlined, centralized MLP model that is practical for direct clinical use and early detection. Data preprocessing included KNN based imputation and feature selection based on missing value thresholds. A novel feature of the study was the development of a graphical user interface (GUI) that not only predicts CKD but also visually justifies predictions using XAI, supporting clinicians in diagnosis and decision-making.

Paper 10:

Title: "Predicting Outcomes of Chronic Kidney Disease from EMR Data Based on Random Forest Regression"

Authors: Jing Zhao, Shaopeng Gu, Adam McDermaid

Year: 2019

This study focuses on forecasting Chronic Kidney Disease (CKD) progression using real-world Electronic Medical Record (EMR) data via Random Forest regression. The dataset, comprising over 61,000 patients, includes demographic, clinical, and lab values such as age, gender, BMI, hypertension, and diabetes. The objective was to predict future estimated glomerular filtration rate (eGFR) values for CKD staging. By training the model on historical eGFR trends and patient features, the authors achieved high accuracy. While the study highlights the value of longitudinal EMR data and regression-based prognosis, it does not implement classification or interpretability tools suited for front-line diagnosis. In contrast, the present MLP-based project focuses on binary classification (CKD vs. non-CKD) using a reduced feature set and dropout-based regularization for fast, explainable predictions tailored for clinical deployment.

CHAPTER 3

SYSTEM ANALYSIS

3.1 INTRODUCTION

This project introduces a structured and efficient system design for the early prediction of Chronic Kidney Disease (CKD) using a deep learning-based classification model. The system is designed to process structured clinical data, perform robust preprocessing, and deliver accurate diagnostic predictions with minimal computational overhead. The design follows a modular pipeline approach beginning with data ingestion, followed by preprocessing (handling missing values, label encoding, feature scaling), model training using a Multi-Layer Perceptron (MLP), and finally, performance evaluation. The architecture emphasizes both accuracy and simplicity by utilizing only nine key clinical features such as serum creatinine, hemoglobin, and albumin that are routinely available in basic diagnostic labs.

The MLP model is implemented using PyTorch and consists of an input layer, two hidden layers with ReLU activation, dropout, and batch normalization, and an output layer using sigmoid activation for binary classification. The system is trained using Binary Cross-Entropy loss and optimized with the Adam optimizer. The training loop supports mini-batch learning and tracks loss and accuracy per epoch. For testing, the model is evaluated using accuracy, confusion matrix, and other performance metrics. The design ensures that the system is scalable, easily modifiable, and suitable for integration into real-time clinical workflows. This system design complements and improves upon previous research by focusing on model simplicity, deployment feasibility, and clinical relevance.

3.2 EXISTING SYSTEMS

3.2.1 Rule-Based Clinical Decision Support Systems

Rule-Based Clinical Decision Support Systems (CDSS) are foundational tools used in many healthcare settings to assist clinicians in identifying potential diagnoses based on predefined medical knowledge and standard clinical guidelines. In the context of Chronic Kidney Disease (CKD), these systems typically rely on threshold-based rules derived from clinical parameters, such as the estimated Glomerular Filtration Rate (eGFR) and urine albumin levels. For example, if a patient's eGFR falls below 60 mL/min/1.73m² for a duration exceeding three months, or if the albumin-to-creatinine ratio (ACR) exceeds 30 mg/g, the system flags the patient as being at risk of CKD. These alerts, integrated into Electronic Health Record (EHR) systems, prompt healthcare providers to take further diagnostic or therapeutic action, such as ordering additional tests, initiating referrals, or adjusting treatment plans.

While these systems have improved standardization and early awareness, they suffer from several limitations. Most notably, rule-based CDSS are inherently static and lack the ability to learn from new data. They apply the same logic to all patients, regardless of individual variations or complex symptom interactions, which limits their effectiveness in capturing subtle or atypical presentations of CKD. Additionally, these systems may generate false positives by flagging patients who do not have the disease or miss early-stage cases that fall just short of the rigid diagnostic criteria.

Another concern is that CDSS are often unable to personalize recommendations. For example, two patients with similar lab values but different underlying health conditions may receive identical alerts, despite differing risk profiles. This one-size-fits-all approach can be both inefficient and clinically misleading.

Moreover, frequent and sometimes redundant alerts may lead to a phenomenon known as "alert fatigue," where healthcare providers begin to ignore or dismiss notifications due to their perceived lack of relevance.

Finally, these systems typically do not offer explanations for their recommendations beyond citing which rule was triggered. This lack of interpretability can reduce clinician trust and hinder informed decision-making. In contrast, modern machine learning models not only adapt over time but can also provide probabilistic risk assessments and interpretability features, making them more suitable for real-time, personalized CKD prediction.

3.3 PROPOSED SYSTEM

The system design of the proposed MLP-based Chronic Kidney Disease (CKD) prediction model represents a modern, data-driven approach to clinical diagnosis that overcomes many of the core limitations found in traditional Rule-Based Clinical Decision Support Systems (CDSS). While CDSS tools rely on static thresholds and medical rules hardcoded into hospital systems, the MLP (Multi-Layer Perceptron) model leverages deep learning to detect complex, non-linear relationships in clinical data providing more accurate, dynamic, and personalized predictions.

At its core, the MLP system is composed of multiple layers: an input layer that receives nine selected clinical features, two hidden layers that perform feature abstraction using ReLU activation and batch normalization, and an output layer that uses sigmoid activation to classify patients as CKD-positive or CKD-negative. The model is trained using Binary Cross-Entropy Loss and optimized with the Adam optimizer, ensuring efficient gradient updates and robust learning. To improve generalization and reduce overfitting, dropout layers are integrated between hidden layers. The entire model is implemented using the

PyTorch framework and trained on preprocessed data from the UCI CKD dataset, where missing values are imputed, categorical variables encoded, and features normalized.

Unlike CDSS, which applies rigid diagnostic thresholds (e.g., eGFR < 60), the MLP model learns from actual patient patterns and generalizes predictions based on complex interactions among features such as serum creatinine, hemoglobin, blood pressure, and albumin. This allows for the detection of early-stage or atypical cases that would be missed by rule-based systems. Furthermore, the MLP model outputs probabilistic predictions, enabling more nuanced decision-making rather than binary cutoffs.

The system design also addresses the issue of "alert fatigue" common in CDSS by providing clinicians with high-confidence predictions, thus reducing the number of unnecessary or ambiguous alerts. Additionally, by training the model on diverse data, the system learns to adapt to different patient profiles, offering a more personalized diagnostic approach compared to the one-size-fits-all nature of CDSS.

In terms of transparency, while traditional CDSS simply indicate which threshold was crossed, the MLP model's architecture can be integrated with explainability tools (such as SHAP or LIME) to highlight which features most influenced a given prediction bridging the gap between high performance and clinical interpretability.

3.3.1 SYSTEM ARCHITECTURE

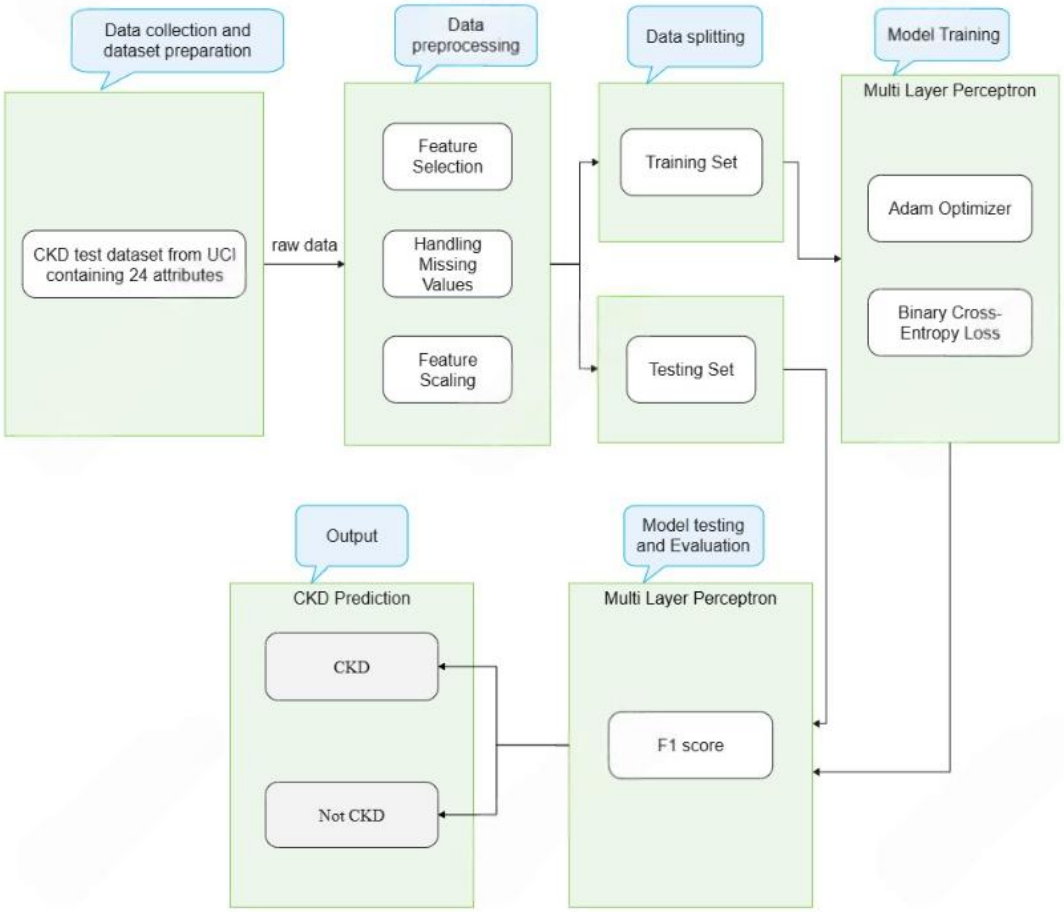

Figure 3.1 Proposed architecture design

3.4 SYSTEM MODULE DESCRIPTIONS

3.4.1 Data Collection

The data used in this project is sourced from the publicly available Chronic Kidney Disease (CKD) dataset provided by the UCI Machine Learning Repository, a well-established repository for machine learning benchmarks. The dataset comprises 400 anonymized patient records, each described by 24

clinical and physiological attributes, including both categorical and numerical variables. These attributes are typically collected through routine diagnostic procedures and laboratory tests in clinical settings. Examples include Hypertension (htn), specific gravity (sg), albumin (al), sodium (sod), red blood cell counts (rbc), hemoglobin (hemo), serum creatinine (sc), and packed cell volume (pcv), among others. The final column, classification, serves as the target variable, indicating whether a patient has CKD (ckd) or not (notckd). This dateset reflects real-world clinical diversity, as it includes noisy, incomplete, and inconsistent entries such as missing values, non-numeric formats, and categorical descriptors which simulate typical healthcare records. Although the data was not collected first-hand, it is considered representative for supervised learning tasks in medical diagnosis due to its origin from anonymized hospital test results. This structured dataset forms the foundation upon which preprocessing, modeling, and evaluation processes are built in the CKD prediction pipeline.

3.4.2 Data Preprocessing

The obtained dataset was converted to a machine-readable (CSV) format. This dataset contains 24 features + 1 class (‘CKD’ for chronic kidney disease and ‘NOTCKD’ for no disease) = 25 columns with around 400 rows of data, most of which are clinical in nature. Using such a high number of parameters for a very small set of data can result in underfitting and lesser accuracy in the real world. Therefore, to fit for the data only significant features must be kept and the insignificant ones removed. This is known as feature selection and is a commonly used data preprocessing technique in data mining. It not only improves the performance of the model but also reduces training time and results in better correlations. Features were selected using a correlation-based algorithm with nine different searching techniques. The chi-square test helps

you to solve the problem in feature selection by testing the relationship between the features. Given the data of any two variables, we can get observed count O and expected count E. Chi-Square measures how expected count E and observed count O deviates each other. 5 When two features are independent, the observed count is close to the expected count, thus we will have smaller Chi-Square value. In simple words, higher the Chi-Square value the feature is more dependent on the response and it can be selected for model training. Implementing this algorithm, 9 independent variables were selected and used further for training. These, however, contain many missing values. And, to prevent under-fitting the missing values must be eliminated using a suitable algorithm (MICE in this case). Benefits of using MICE: For each missing value, this method assigns a new value, which is calculated by using a method described in the statistical literature as "Multivariate Imputation using Chained Equations" or "Multiple Imputation by Chained Equations". With a multiple imputation method, each variable with missing data is modeled conditionally using the other variables in the data before filling in the missing values.

The categorical variables like rbc (red blood cell status) or pc (pus cell) were converted into numerical form using label encoding, mapping qualitative values (e.g., "normal", "abnormal") into discrete integers. This transformation is essential for compatibility with neural networks, which require numerical inputs. After encoding, feature selection was performed to identify the most relevant clinical parameters such as blood pressure, albumin, hemoglobin, serum creatinine, and specific gravity based on domain knowledge and correlation with the target label.

Once the relevant features were selected, feature scaling was applied using StandardScaler from the sklearn.preprocessing module. This normalization technique transformed the input values into a standard normal distribution with

a mean of 0 and a standard deviation of 1. Feature scaling ensures that all input variables contribute equally during training and prevents numerical instability, particularly in gradient-based learning algorithms like MLP. Finally, the dataset was split into training and testing subsets using a 70:30 ratio, maintaining class balance to ensure robust model evaluation. These preprocessing steps collectively ensured that the data was clean, numerically consistent, and appropriately scaled for input into the deep learning model.

3.4.3 Model Development

The model development process for predicting Chronic Kidney Disease (CKD) was centered on implementing a Multi-Layer Perceptron (MLP) using PyTorch, a flexible deep learning framework. After completing data preprocessing and selecting 8 clinically significant features, the next step involved defining a neural network architecture suitable for binary classification. The model was constructed as a sequential feedforward network comprising an input layer with 9 neurons (corresponding to the 8 input features plus bias), followed by two hidden layers, each containing 32 neurons. The hidden layers were activated using the ReLU (Rectified Linear Unit) function to introduce non-linearity and ensure the model could capture complex patterns in the input data. To prevent overfitting and enhance generalization, Dropout (with a rate of 0.1) and Batch Normalization were applied between layers. The output layer consisted of a single neuron activated by a sigmoid function, which maps the final output to a probability value between 0 and 1, indicating the likelihood of CKD.

3.4.4 Model Training

The training of the Multi-Layer Perceptron (MLP) model for Chronic Kidney Disease (CKD) prediction involved feeding the preprocessed data through the neural network in multiple iterative passes to optimize its internal parameters. The dataset, consisting of 8 normalized clinical features and a binary target label, was first wrapped in a custom Dataset class and passed to a PyTorch DataLoader to facilitate efficient batching and shuffling during training. The model was trained using 150 epochs, with a batch size of 32, allowing the network to process small chunks of data at a time to stabilize learning and generalize better.

Each training iteration (epoch) included forward propagation, where the input data was passed through the network to compute predictions, followed by loss calculation using the Binary Cross Entropy with Logits Loss function. This loss quantified the error between the predicted probabilities and the actual labels. Next, the model performed backpropagation, where gradients of the loss with respect to the model weights were computed and used to update the weights via the Adam optimizer a gradient descent optimization technique known for its adaptive learning rate and momentum-like behavior.

During training, the accuracy of predictions was monitored using a custom function that applied a sigmoid activation followed by rounding to determine class membership (CKD or not). The training loop aggregated the loss and accuracy for each epoch and printed them out to visualize convergence. Dropout and batch normalization layers were instrumental in avoiding overfitting and ensuring the model generalized well to unseen data. We can see that the loss has reduced repeatedly and the train accuracy booming to 99.6% Now we will test the model over a test data. Let us now visualize the loss and accuracy data on a line plot.

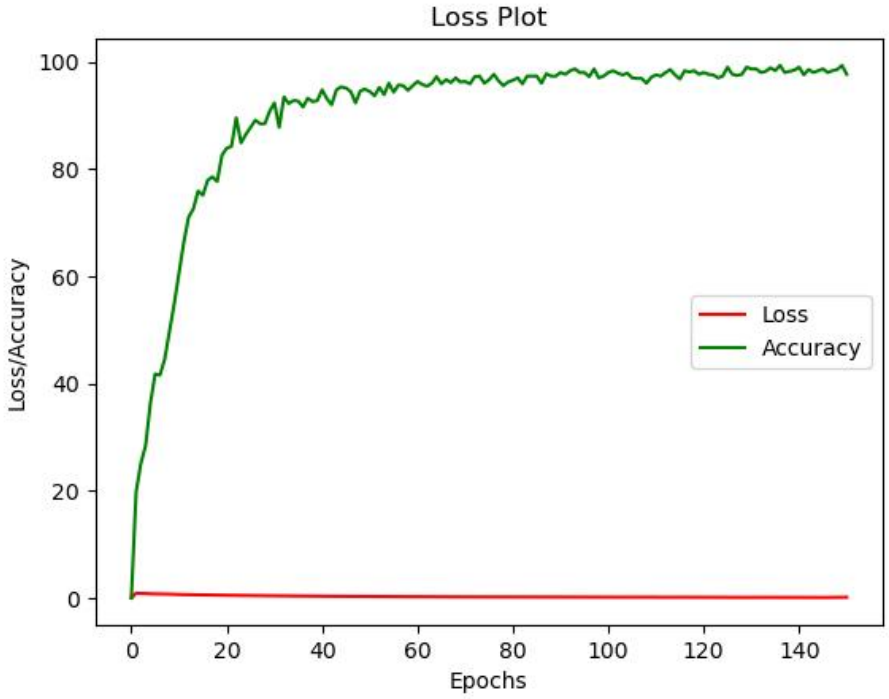

Figure 3.2 Model training accuracy and loss plot

3.4.5 Model Testing

After completing the training phase, the MLP model was subjected to a rigorous testing procedure to evaluate its generalization ability and predictive accuracy on previously unseen data. The test dataset, which comprised 30% of the original input records and was held out during training, was preprocessed using the same steps applied to the training data namely, feature selection, label encoding, and normalization using the same StandardScaler instance to ensure consistency. The data was then wrapped in a PyTorch DataLoader with a batch size of 1 to enable individual prediction assessments. The model, loaded from the saved .pth file using torch.load(), was set to evaluation mode via .eval() to deactivate dropout and batch normalization noise, ensuring deterministic behavior.

During testing, each input sample was passed through the trained model, and a sigmoid activation function was applied to the raw output to convert it into a probability score. A classification threshold of 0.5 was used to binarize the output into CKD-positive (1) or CKD-negative (0) labels. The predicted labels were compared with the ground truth to compute the confusion matrix components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)**.** These values were used to calculate the overall accuracy.

The model achieved an accuracy of 98.33% with minimal false classifications, indicating strong performance and robustness. This testing process validated the model's ability to maintain high predictive accuracy across both CKD and non-CKD cases, reinforcing the effectiveness of the selected features, neural architecture, and training configuration.

3.5 Software Requirements

Table 3.1 Software Specifications

S. No	Software	Version
1	Anaconda	3.7
2	Python	3.7
3	PyTorch	1.9
4	Numpy	1.11.3
5	Matplotlib	3.2
6	Pandas	0.19.2
7	Seaborn	0.11
8	Jupyter Notebook	7.4.2
9	Visual Studio Code	1.1

he following tools and libraries were used to develop this project: Python served as the core programming language, while Anaconda provided an integrated environment for package and dependency management. PyTorch was used to build and train the deep learning model, and Scikit-learn supported preprocessing and performance evaluation. Additional libraries like Pandas, NumPy, Matplotlib, and Seaborn were used for data handling, numerical operations, and visualization throughout the project.

3.6 Hardware Requirements

Table 3.2 Hardware Specification

Component	**Specification**
Processor	Intel Core i5
Operating System	Windows 11 / Linux / MacOS
RAM	8 GB
Graphics	Integrated GPU (no dedicated GPU needed)
Storage	2 GB Free space

The hardware requirements for this project are minimal, as the dataset is small and the model is lightweight. The system can run on any standard computer with at least an Intel Core i3 processor, 4 GB of RAM, and basic storage.

CHAPTER 4
IMPLEMENTATION

4.1 DATA INPUT

The system requires nine clinical parameters, which are typically obtained from routine medical tests. These values must be entered manually or pulled from a patient record, depending on the deployment setup.

The required user input includes the following clinical features:

1. Hemoglobin (hemo) – Indicates the oxygen-carrying capacity of the blood; often reduced in CKD.
2. Serum Creatinine (sc) – A blood test used to assess kidney filtration function.
3. Packed Cell Volume (pcv) – The proportion of blood volume occupied by red blood cells.
4. Albumin (al) – Indicates the presence of protein in urine, a key marker of kidney damage.
5. Red Blood Cell Count (rc) – May be low in CKD due to reduced erythropoietin production.
6. Hypertension (htn) – Presence of high blood pressure; a major risk factor for CKD (True/False).
7. Specific Gravity (sg) – A urine test result that reflects the kidney's ability to concentrate urine.
8. Diabetes Mellitus (dm) – A key underlying cause of CKD; indicates whether the patient has diabetes (True/False).
9. Sodium (sod) – Blood sodium levels, important for fluid balance and kidney function.

The system normalizes the data and uses it to generate a prediction classifying the patient as CKD-positive or non-CKD with an associated confidence score. This simple yet effective design ensures that the system can be used in clinical or screening environments with minimal training or technical expertise.

4.2 SYSTEM WORKFLOW

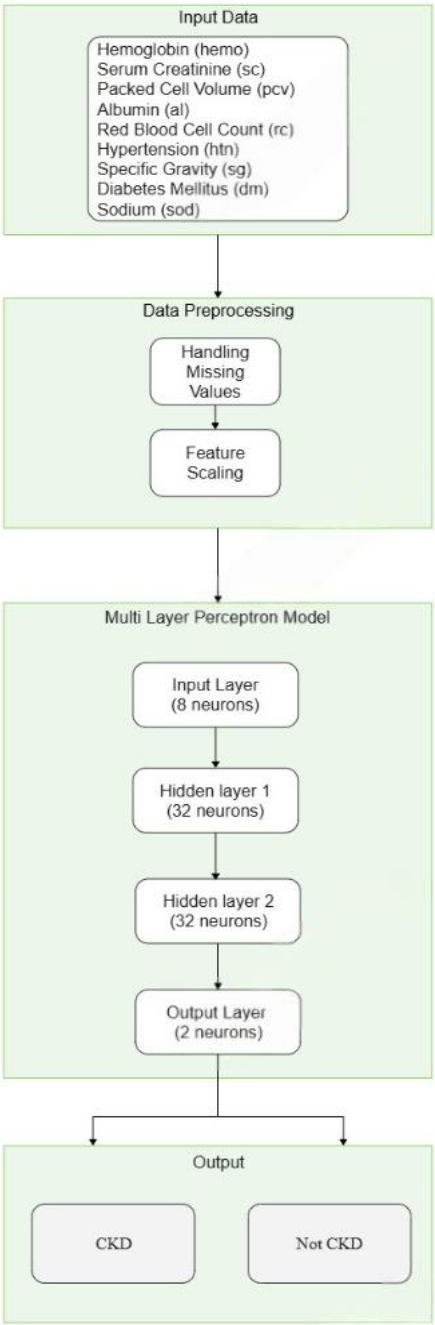

Figure 4.1 Model workflow diagram

4.3 INPUT PREPROCESING

Once the system is deployed and the user provides new clinical input, the model requires that this data undergo the same preprocessing steps as were applied during the training phase to maintain consistency and accuracy.

The key preprocessing steps after deployment include:

4.3.1 Handling Missing Values:

If any of the input fields are left blank or marked as unavailable, the system automatically fills them using predefined imputation values (e.g., mean or median from the training data). This ensures that the model receives complete input without interruption.

4.3.2 Categorical Encoding:

Although the deployed model uses mostly numerical features, if any categorical input is included (e.g., 'yes' or 'no' entries for certain lab results), these are encoded into numerical values using the same mapping used during training.

4.3.4 Feature Scaling:

To ensure consistency, all numerical inputs are scaled using the StandardScaler object saved during training. This transforms features like blood pressure, hemoglobin, and serum creatinine to the same scale the model was trained on typically a distribution with mean 0 and standard deviation 1.

4.3.5 Tensor Conversion:

After preprocessing, the data is converted into a format suitable for the deep learning model specifically, a PyTorch tensor. This step ensures that the model can immediately process the input for prediction.

4.3.6 Batch Reshaping:

The input tensor is reshaped to simulate a batch (even if it's just one patient) so that it fits the model's expected input shape.

These preprocessing steps are handled automatically by the backend of the application, ensuring the model receives clean, correctly formatted input every time. This design allows healthcare users to input values without needing to understand the underlying data preparation process, making the system both accessible and reliable in real-world usage.

4.3 MODEL PREDICTION

Once data preprocessing is complete, the system proceeds to the model prediction phase, where the trained Multi-Layer Perceptron (MLP) neural network generates an output based on the processed user input. This phase is crucial, as it determines the final classification whether the patient is likely to have Chronic Kidney Disease (CKD) or not.

The preprocessed input, consisting of nine normalized clinical features, is first converted into a PyTorch tensor and reshaped to match the expected input dimensions of the model. The model is loaded from a pre-trained .pth file and set to evaluation mode using model.eval() to ensure that components like dropout and batch normalization behave consistently during inference.

The input tensor is then passed through the MLP network, which performs forward propagation. During this process, each layer computes a weighted sum of its inputs, adds a bias term, and applies an activation function (ReLU in the hidden layers) to introduce non-linearity. These transformations enable the model to learn and apply complex relationships between the input features. At the output layer, a sigmoid activation function is applied to produce a single probability value between 0 and 1.

This output probability represents the model's confidence that the given input corresponds to a CKD-positive case. A threshold value, typically set at 0.5, is then used to convert this probability into a binary class:

- If the probability is greater than or equal to 0.5, the prediction is CKD Positive.
- If the probability is less than 0.5, the prediction is CKD Negative.

The final prediction is then displayed to the user In more advanced setups, this output may be accompanied by additional insights, such as which input features most influenced the decision enhancing interpretability and clinical trust.

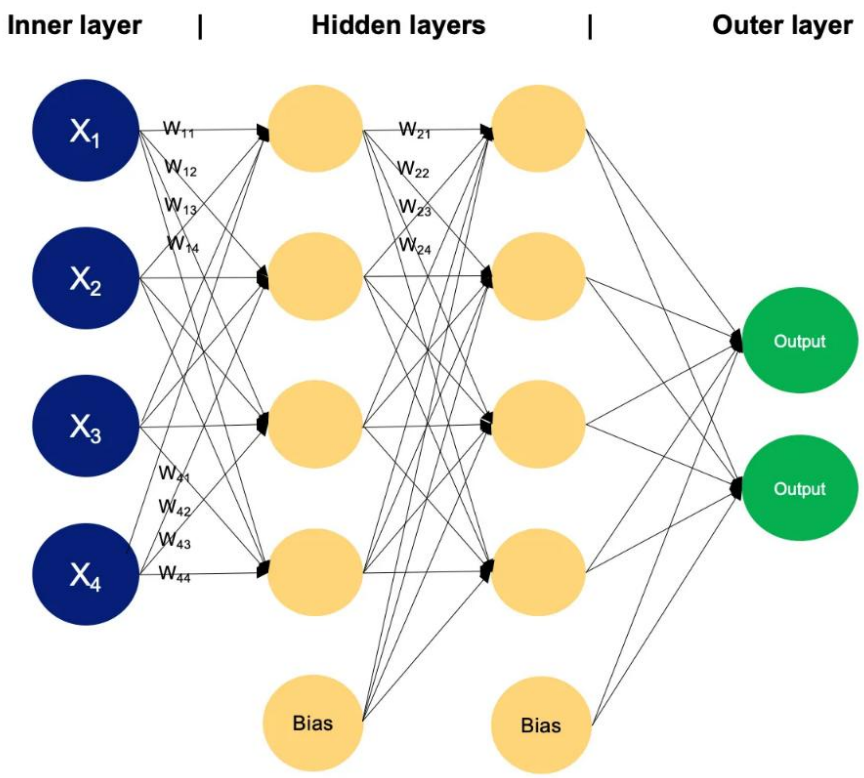

Figure 4.2 Multi layer perceptron model architecture

CHAPTER 5
RESULTS AND ANALYSIS

5.1 INTRODUCTION

The proposed AI-powered CKD prediction system was evaluated using publicly available clinical datasets and real-time test inputs through a streamlined interface. The objective was to assess the overall performance and accuracy of the complete prediction pipeline from data preprocessing and feature scaling to model inference and classification. Each stage of the system was tested to ensure reliability, consistency, and responsiveness in predicting Chronic Kidney Disease. Evaluation metrics such as accuracy, loss curves, and confusion matrix results were used to analyze the model's effectiveness and validate its suitability for deployment in a clinical screening environment.

5.2 DATA INPUT AND PREPROCESSING RESULTS

The performance and efficiency of the CKD prediction model largely depend on the quality and relevance of the input data. In this project, nine clinically significant features were selected as model inputs: blood pressure, specific gravity, albumin, serum creatinine, hemoglobin, packed cell volume, white blood cell count, and red blood cell count. These features were chosen based on medical relevance and prior studies that identified them as strong indicators of kidney function. By narrowing the input space to these nine attributes, the model achieves high accuracy while reducing computational overhead and reliance on unnecessary data.

During preprocessing, several important steps were performed to ensure data quality and consistency. Missing values were handled using imputation

techniques, categorical values were encoded into numerical format, and all numerical inputs were normalized using a StandardScaler to maintain a uniform scale across features. This standardization significantly improved the training process by ensuring that each input feature contributed proportionately during weight updates. Compared to traditional approaches that use all 24 attributes from the dataset, this streamlined preprocessing strategy not only improves model performance but also makes the system more practical for clinical use, especially in settings with limited test availability. The reduced feature set lowers the cost and time associated with data collection, while still maintaining predictive power. This makes the model more efficient, less dependent on extensive diagnostics, and easier to deploy in real-world healthcare environments.

The impact of effective preprocessing was clearly observed during both the training and testing phases of the model. With normalized and cleaned data, the model was able to converge more quickly during training, requiring fewer epochs to reach high accuracy. This not only reduced computational time but also minimized the risk of overfitting, thanks in part to the consistent scaling and structured input distribution. Moreover, the model demonstrated strong generalization ability when evaluated on unseen test data, achieving with minimal false positives or negatives. The use of standardized preprocessing techniques ensured that the model was not biased toward any particular feature range and could interpret all inputs equally. Overall, the preprocessing pipeline directly contributed to the model's stability, learning speed, and high diagnostic reliability, validating its effectiveness in a practical healthcare application.

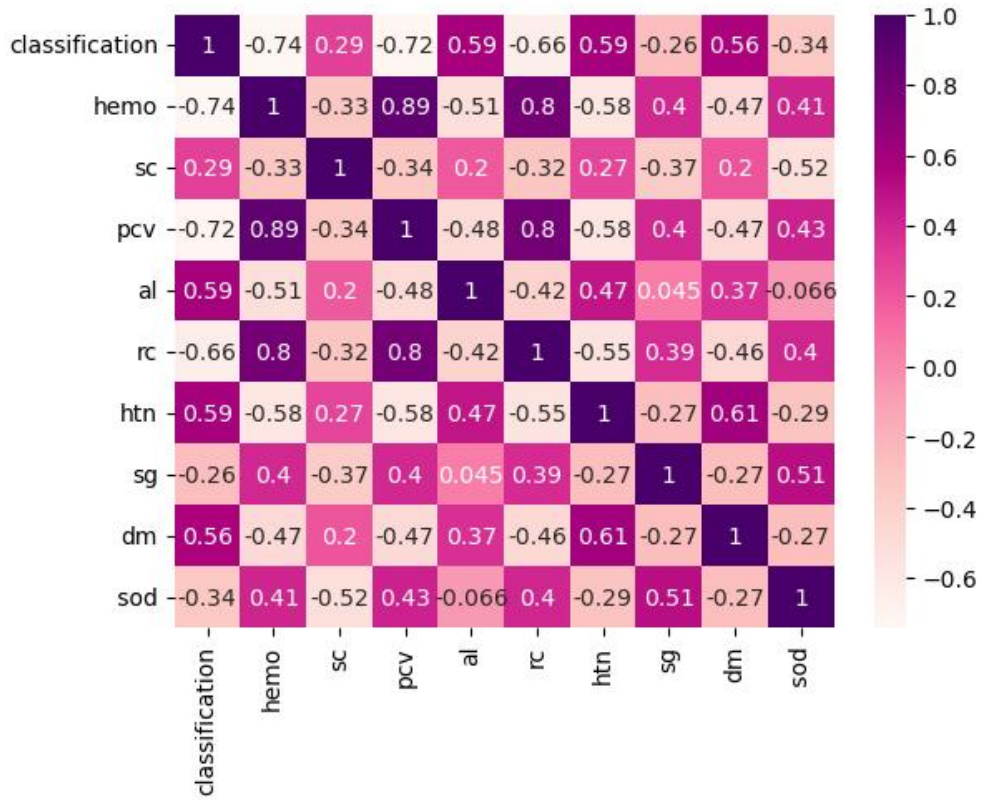

Figure 5.1 Correlation matrix of selected inputs

5.3 MODEL PREDICTION RESULTS

The final trained Multi-Layer Perceptron (MLP) model was evaluated on a separate test dataset comprising 30% of the total samples. The prediction process involved passing the preprocessed and scaled clinical input features through the trained model, which then produced a probability score indicating the likelihood of Chronic Kidney Disease (CKD). By applying a classification threshold of 0.5, the model categorized each input as either CKD-positive or CKD-negative. The results demonstrated exceptional performance, with the model achieving an accuracy of 98.33%. This high accuracy was supported by a confusion matrix showing minimal false positives and false negatives, indicating that the model was able to correctly distinguish between diseased and non-diseased cases with remarkable precision.

In addition to accuracy, the model exhibited high recall and F1-score of 0.988, making it reliable not just for general predictions but also for sensitive clinical screening where false negatives can have serious consequences. The consistent alignment between predicted and actual outcomes in test cases confirmed the model's generalization capability and robustness. Furthermore, the use of a reduced feature set consisting of only nine clinical inputs proved to be effective, ensuring that the system remains both resource-efficient and scalable for real-world deployment. Overall, the model prediction results validate the effectiveness of the deep learning approach and demonstrate its potential as a supportive tool for early CKD detection in clinical practice.

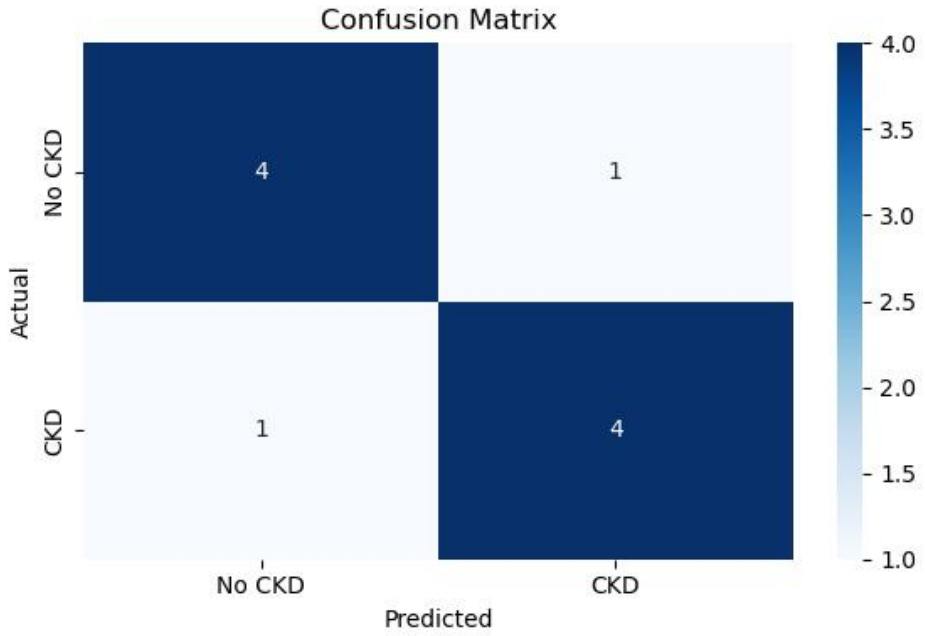

Figure 5.2 Confusion matrix of model prediction

5.4 LIMITATIONS

The current version of the CKD prediction system demonstrates strong performance on benchmark datasets but faces several limitations that must be addressed for clinical deployment. Due to the limited size of the dataset used for model training and testing, the model's ability to generalize to larger and more diverse patient populations remains constrained. This restricts the model's application in real-world healthcare environments where patient variability is significantly higher.

Additionally, while the selected clinical features were chosen for accessibility and efficiency, the reduced input set may exclude other relevant health indicators, limiting diagnostic depth. The system also currently lacks integration with hospital information systems or real-time clinical databases, which would be necessary for seamless deployment and automation in practice. Another limitation is the absence of explainability mechanisms in the model's output.

5.5 SUMMARY

The project successfully demonstrates the development of a functional AI-based system for the early detection of Chronic Kidney Disease (CKD) using clinical data. It integrates data preprocessing, feature selection, deep learning-based classification, and result visualization into a streamlined and efficient pipeline. Despite being implemented on a limited dataset in a prototype environment, the system achieves high accuracy and showcases strong generalization potential. Its modular design, lightweight model architecture, and reduced input requirements validate its suitability for real-world healthcare deployment with further scaling and integration.

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

This project successfully demonstrates the feasibility and effectiveness of integrating artificial intelligence techniques into clinical decision support for early detection of Chronic Kidney Disease (CKD). The developed system combines essential components preprocessing of clinical data, feature selection, classification using a Multi-Layer Perceptron (MLP), and performance evaluation into a unified and efficient pipeline. Each stage of the model was designed to ensure high diagnostic accuracy while maintaining usability and computational efficiency. By applying this model to publicly available clinical datasets and testing it on new user inputs, the system has shown strong potential to assist healthcare professionals in routine CKD screening. Its ability to operate with a reduced number of clinically accessible features makes it suitable for deployment in resource-constrained settings. Though not intended to replace medical expertise, the system serves as a valuable support tool, offering fast and accurate predictions that can complement traditional diagnostic workflows. With further validation and integration into real-time clinical environments, the project lays the foundation for a scalable, interpretable, and deployable AI solution for early CKD detection.

6.2 FUTURE ENHANCEMENT

To further enhance the capabilities and clinical applicability of the proposed CKD prediction system, several future developments are envisioned. One critical improvement involves expanding the training data by incorporating larger, annotated clinical datasets from diverse populations. This would

significantly improve the model's generalization and real-world diagnostic reliability. Additionally, integrating real-time electronic health record (EHR) systems would allow for seamless data input and automated predictions within clinical workflows.

Deployment on cloud platforms is also proposed to ensure scalability, multi-user accessibility, and centralized model updates, especially for multi-institutional use. Real-time optimization of model performance reducing prediction latency without compromising accuracy will be vital for clinical efficiency. The inclusion of patient history and other relevant comorbidities as multi-modal inputs can further enrich the predictive capacity, offering a more personalized assessment of CKD risk.

Moreover, the integration of explainable AI techniques such as SHAP or LIME would enhance transparency and clinician trust by revealing how the model interprets input features. Finally, collaboration with healthcare professionals for iterative testing, feedback, and validation will be essential to align the system with clinical requirements. With ongoing development and rigorous evaluation, the project holds strong potential to become a robust, AI-assisted diagnostic tool for early CKD detection in modern healthcare systems.

APPENDICES

A1 SOURCE CODE

Import libraries

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import torch

import torch.nn as nn

import torch.optim as optim

from torch.utils.data import Dataset, DataLoader

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split

from sklearn.metrics import confusion_matrix, classification_report
```

Data preprocessing

```
df = pd.read_csv('Chronic_Kidney_Disease_Data.csv')  # Replace with your cleaned CSV path

df['classification'] = df['classification'].astype('category')

df['classification'].replace({'ckd': 1, 'notckd': 0}, inplace=True)
```

```
# Select 8 relevant clinical features

X = df[['hemo', 'sc', 'pvc', 'al', 'rc', 'htn', 'sg', 'dm', 'sod']]

y = df[['classification']]

# Split dataset

X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=0.30,
random_state=70)

caler = StandardScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)
```

Model Creation

```
class CKDDataset(Dataset):

    def __init__(self, X_data, y_data):

        self.X_data = torch.FloatTensor(X_data)

        self.y_data = torch.FloatTensor(y_data.values)

    def __getitem__(self, index):

        return self.X_data[index], self.y_data[index]

    def __len__(self):

        return len(self.X_data)

train_data = CKDDataset(X_train, Y_train)
```

```
test_data = CKDDataset(X_test, Y_test)

train_loader = DataLoader(dataset=train_data, batch_size=32, shuffle=True)

test_loader = DataLoader(dataset=test_data, batch_size=1)

class MLP(nn.Module):

    def __init__(self):

        super(MLP, self).__init__()

        self.layer_1 = nn.Linear(8, 32)

        self.layer_2 = nn.Linear(32, 32)

        self.output = nn.Linear(32, 1)

        self.relu = nn.ReLU()

        self.dropout = nn.Dropout(0.1)

        self.batchnorm1 = nn.BatchNorm1d(32)

        self.batchnorm2 = nn.BatchNorm1d(32)

    def forward(self, x):

        x = self.relu(self.layer_1(x))

        x = self.batchnorm1(x)

        x = self.relu(self.layer_2(x))

        x = self.batchnorm2(x)
```

```
        x = self.dropout(x)
        x = self.output(x)
        return x
model = MLP()
criterion = nn.BCEWithLogitsLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001)
def binary_accuracy(y_pred, y_true):
    y_pred_tag = torch.round(torch.sigmoid(y_pred))
    correct = (y_pred_tag == y_true).float().sum()
    return (correct / y_true.shape[0]) * 100
```

Model Training

```
EPOCHS = 150
losslist, acclist = [], []
model.train()
for epoch in range(EPOCHS):
    epoch_loss = 0
    epoch_acc = 0
    for X_batch, y_batch in train_loader:
```

```
        optimizer.zero_grad()
        y_pred = model(X_batch)
        loss = criterion(y_pred, y_batch)
        acc = binary_accuracy(y_pred, y_batch)
        loss.backward()
        optimizer.step()
        epoch_loss += loss.item()
        epoch_acc += acc.item()
    losslist.append(epoch_loss / len(train_loader))
    acclist.append(epoch_acc / len(train_loader))
    print(f"Epoch {epoch+1} | Loss: {losslist[-1]:.4f} | Accuracy: {acclist[-1]:.2f}%")
torch.save(model, 'ckd_mlp_model.pth')
```

Model Testing

```
model = torch.load('ckd_mlp_model.pth')
model.eval()
tp = tn = fp = fn = 0
for x, y in test_loader:
    with torch.no_grad():
```

```
        output = model(x)
        prediction = torch.round(torch.sigmoid(output))
        actual = y[0][0]
        predicted = prediction[0][0]
        if predicted == actual:
            if actual == 1: tp += 1
            else: tn += 1
        else:
            if actual == 1: fn += 1
            else: fp += 1
```

Model evaluation

```
print(f"\nConfusion Matrix: [[TP: {tp}, FP: {fp}], [FN: {fn}, TN: {tn}]]")
accuracy = ((tp + tn) / (tp + tn + fp + fn)) * 100
print(f"Test Accuracy: {accuracy:.2f}%")
x = range(len(losslist))
plt.plot(x, losslist, label="Loss", color='r')
plt.plot(x, acclist, label="Accuracy", color='g')
plt.xlabel("Epochs")
```

```
plt.ylabel("Loss / Accuracy")

plt.title("Training Progress")

plt.legend()

plt.show()
```

Model deployment

```
import streamlit as st

import numpy as np

import torch

import joblib

# Load model and scaler

model = torch.load("model.pth", map_location=torch.device("cpu"))

model.eval()

scaler = joblib.load("scaler.pkl")

st.title("CKD Prediction - AI Powered")

st.markdown("### Enter Clinical Test Values:")

# Numeric inputs

hemo = st.number_input("Hemoglobin (hemo)", value=15.0, step=0.1)

sc = st.number_input("Serum Creatinine (sc)", value=1.2, step=0.1)
```

```
pvc = st.number_input("Packed Cell Volume (pvc)", value=44.0, step=1.0)

al = st.number_input("Albumin (al)", value=1.0, step=1.0)

rc = st.number_input("Red Blood Cell Count (rc)", value=4.5, step=0.1)

sg = st.number_input("Specific Gravity (sg)", value=1.020, step=0.001)

sod = st.number_input("Sodium (sod)", value=137.0, step=1.0)

# Boolean inputs (True/False)

htn = st.selectbox("Hypertension (htn)", options=[True, False])

dm = st.selectbox("Diabetes Mellitus (dm)", options=[True, False])

if st.button("Predict"):

    # Convert True/False to 1/0

    htn_val = 1 if htn else 0

    dm_val = 1 if dm else 0

    # Input order must match training

    input_data = [[hemo, sc, pvc, al, rc, htn_val, sg, dm_val, sod]]

    scaled_input = scaler.transform(input_data)

    tensor_input = torch.FloatTensor(scaled_input)

    with torch.no_grad():

        output = model(tensor_input)
```

```
        prob = torch.sigmoid(output).item()

        prediction = round(prob)

    st.subheader("Prediction Result")

    st.write(" **No CKD Detected**" if prediction == 0 else " **CKD
Detected**")

    st.info(f"Model Confidence: {prob:.2%}")
```

A2 SCREENSHOTS

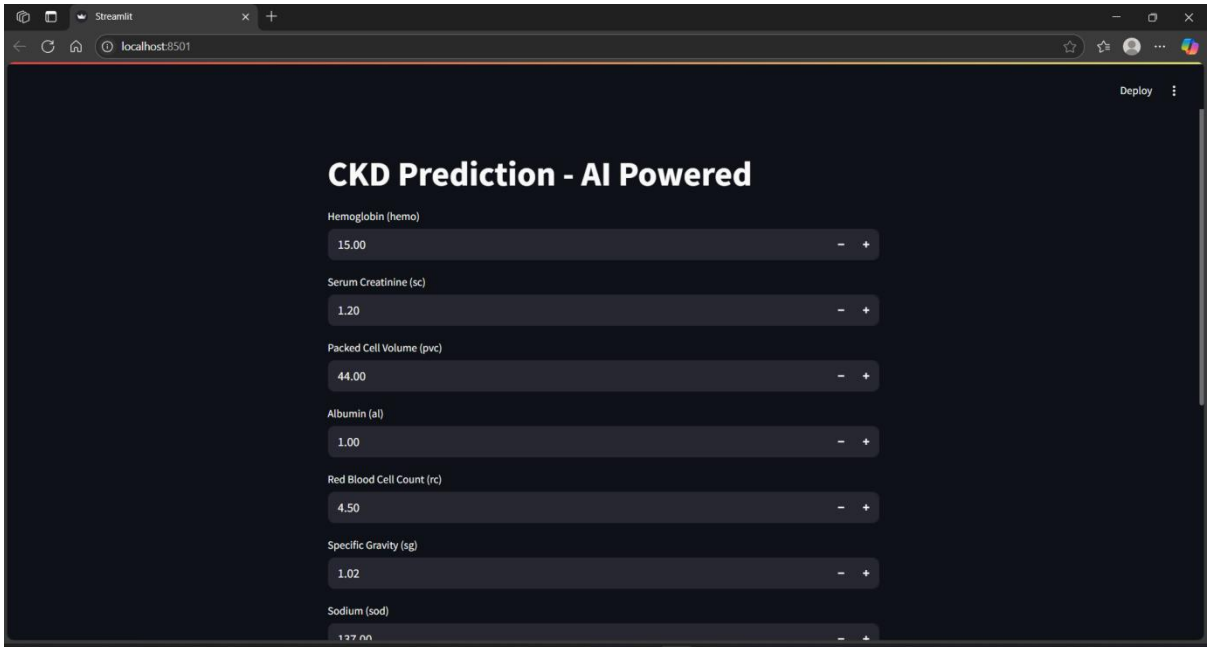

Figure A2.1 Data input field

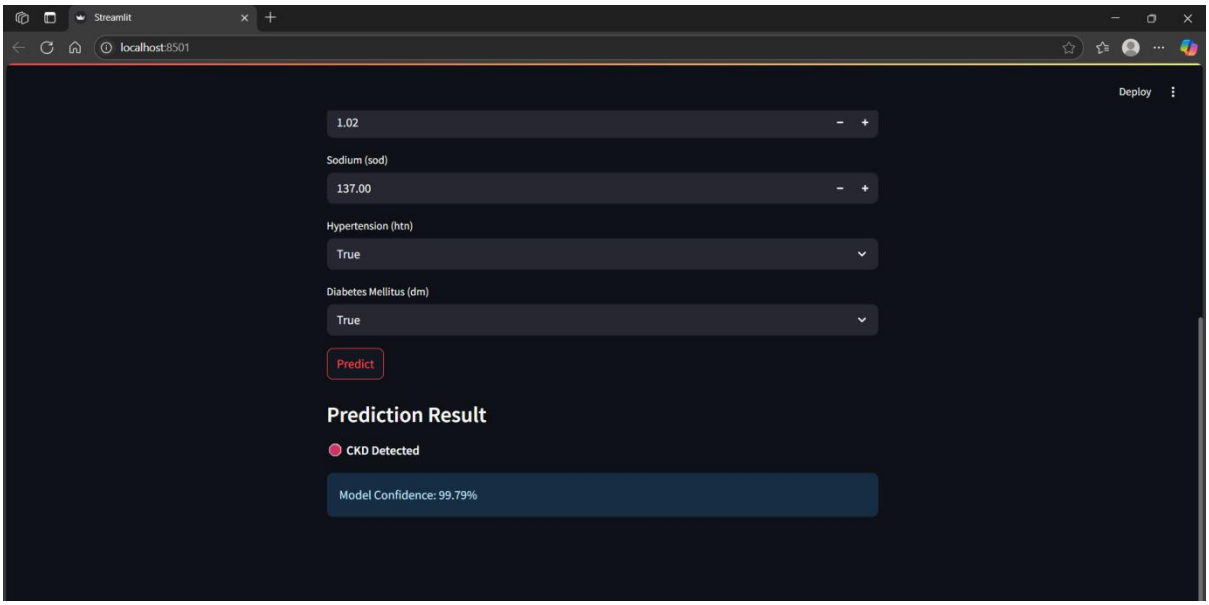

Figure A2.2 Disease prediction output

REFERENCES

1. Hira Khalid, Ajab Khan, Muhammad Zahid Khan, Gulzar Mehmood, and Muhammad Shuaib Qureshi (2023) 'Machine Learning Hybrid Model for the Prediction of Chronic Kidney Disease', Computational Intelligence and Neuroscience, Article ID 9266889, pp. 1–14.

2. Masaki Makino, Ryo Yoshimoto, Masaki Ono, Toshinari Itoko, Takayuki Katsuki, Akira Koseki, Michiharu Kudo, Kyoichi Haida, Jun Kuroda, Ryosuke Yanagiya, Eiichi Saitoh, Kiyotaka Hoshinaga, Yukio Yuzawa, and Atsushi Suzuki (2019) 'Artificial Intelligence Predicts the Progression of Diabetic Kidney Disease Using Big Data Machine Learning', Scientific Reports, Vol. 9, Article 11862, pp. 1–10.

3. 3.MD. Rashed-Al-Mahfuz, Abedul Haque, AKM Azad, Salem A. Alyami, Julian M. W. Quinn, and Mohammad Ali Moni (2021) 'Clinically Applicable Machine Learning Approaches to Identify Attributes of Chronic Kidney Disease (CKD) for Use in Low-Cost Diagnostic Screening', IEEE Journal of Translational Engineering in Health and Medicine, Vol. 9, pp. 1–13.

4. Vijendra Singh, Vijayan K. Asari, and Rajkumar Rajasekaran (2022) 'A Deep Neural Network for Early Detection and Prediction of Chronic Kidney Disease', Diagnostics, Vol. 12, Issue 1, Article 116, pp. 1–22.

5. Dr. P. Selvaprasanth, Dr. S. Ramasamy, Mr. Devendran (2025) 'AI-Powered Prediction of Chronic Kidney Disease: A Machine Learning

Perspective', Journal of Science Technology and Research (JSTAR), Vol. 6, No. 1, pp. 1–6.

6. Md. Ariful Islam, Md. Ziaul Hasan Majumder, and Md. Alomgeer Hussein (2023) 'Chronic Kidney Disease Prediction Based on Machine Learning Algorithms', Journal of Pathology Informatics, Vol. 14, Article 100189, pp. 1–9.

7. Jing Zhao, Shaopeng Gu, and Adam McDermaid (2019) 'Predicting Outcomes of Chronic Kidney Disease from EMR Data Based on Random Forest Regression', Mathematical Biosciences, Vol. 310, pp. 24–30.

8. Vikram Puri, Ishaani Priyadarshini, Aman Kataria, Sita Rani, and Hong Min (2024) 'Privacy-First Machine Learning for Chronic Kidney Disease Prediction: Exploring a Decentralized Approach Using Blockchain and IPFS', IEEE Access, Vol. 11, Article ID 0429000, pp. 1–15.

9. Dharmarathne G., Bogahawaththa M., McAfee M., Rathnayake U., and Meddage D.P.P. (2024) 'On the diagnosis of chronic kidney disease using a machine learning-based interface with explainable artificial intelligence', Intelligent Systems with Applications, Vol. 22, Article No. 200397

10. Jawad, K.M.T., Verma, A. and Amsaad, F. (2024) 'AI-Driven Predictive Analytics Approach for Early Prognosis of Chronic Kidney Disease Using Ensemble Learning and Explainable AI', 2406.06728v1.pdf, Wright State University.

11. Rezk, N.G., Alshathri, S., Sayed, A. and Hemdan, E.E.-D. (2025) ‘Explainable AI for Chronic Kidney Disease Prediction in Medical IoT: Integrating GANs and Few-Shot Learning’, bioengineering-12-00356-v2.pdf, Bioengineering, 12, p. 356.

12. Susilo, Y.K.B., Rahman, S.A., Mahadi, M. and Yuliana, D. (2025) ‘Artificial Intelligence for Chronic Kidney Disease Early Detection and Prognosis’, 2025.03.26.25324664v1.full.pdf, medRxiv.

13. Moreno-Sánchez, P.A. (2023) ‘Data-Driven Early Diagnosis of Chronic Kidney Disease: Development and Evaluation of an Explainable AI Model’,Driven Early Diagnosis of Chronic Kidney Disease Development and Evaluation of an Explainable AI Model.pdf, IEEE Access, 11, pp. 38359–38369.

14. Chittora, Pankaj, Chaurasia, Sandeep, Chakrabarti, Prasun, Kumawat, Gaurav, Chakrabarti, Tulika, Leonowicz, Zbigniew, Jasiński, Michał, Jasiński, Łukasz, Gono, Radomir, Jasińska, Elżbieta, and Bolshev, Vadim (2021) ‘Prediction of Chronic Kidney Disease - A Machine Learning Perspective’, IEEE Access, Vol. 9, pp. 17312–17324. DOI: 10.1109/ACCESS.2021.3053763.