Enhanced Chronic Kidney Disease Detection Via Eurygasters Optimization and Ensemble Learning

A Project Report submitted in the partial fulfilment of the requirement for the Award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Bobbepalli Aravind babu	(21471A05L9)
Repudi Vivek	(21471A05P8)
Siddela Vineeth	(22475A0513)

Under the esteemed guidance of

Dodda Venkata Reddy M.Tech.,(Ph.D),

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERINGCOLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A++ Grade and NBA under Tyre -1

NIRF rank in the band of 201- 300 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2024-2025

NARASARAOPETA ENGINEERINGCOLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name **“Enhanced Chronic kidney Disease Detection Via Eurygasters Optimization and Ensemble Learning”** is a bonafide work done by the team B.Aravind babu (21471A05L9), R.Vivek (21471A05P8), S.Vineeth (22475A0513) in partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING
during 2024-2025.

PROJECT GUIDE

Dodda Venkata Reddy M.Tech., (Ph.D)
Assistant Professor

PROJECT CO -ORDINATOR

Dodda Venkata Reddy M.Tech., (Ph.D)
Assistant Professor

HEAD OFTHE DEPARTMENT

Dr. S.N.Tirumala Rao M.Tech., Ph.D
Professor & HOD of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express my thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman **Sri M. V.Koteswara Rao, B.Sc.,**who took keen interest in us in every effort throughout thiscourse. We owe out sincere gratitude to our beloved principal **Dr. S. Venkateswarlu, M.Tech, Ph.D.** for showing his kind attention and valuable guidance throughout the course.

We express my deep felt gratitude towards **Dr. S. N. Tirumala Rao, M.Tech.,Ph.D**, HOD of CSE department and also to my guide **Dodda Venkata Reddy, M.Tech, (Ph.D),** of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend my sincere thanks towards **Dodda Venkata Reddy, M.Tech, (Ph.D),**Associate professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this work. We extend my sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents. We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

Bobbepalli Aravind babu	**(22475A0513)**
Repudi Vivek	**(22475A0513)**
Siddela Vineeth	**(22475A0513)**

DECLARATION

We declare that this project work titled **“Enhanced Chronic kidney Disease Detection Via Eurygasters Optimization and Ensemble Learning”** is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has been submitted for any other degree or professional qualification except as specified.

By

Bobbepalli Aravind babu	**(22475A0513)**
Repudi Vivek	**(22475A0513)**
Siddela Vineeth	**(22475A0513)**

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong leasrning among students with a sense of societal and ethical responsibilities.

Program Specific Outcomes (PSO’s)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, andsynthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineeringactivitieswith an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and giveand receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Project Course Outcomes (CO’S):

CO421.1: Analyse the System of Examinations and identify the problem.
CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.
CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		✓											✓		
C421.2	✓		✓		✓								✓		
C421.3				✓		✓	✓	✓					✓		
C421.4			✓			✓	✓	✓					✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course fro which principles are applied i this project	Description of the device	Attained PO
C2204.2, C22L3.2	Gathering the requirements an defining the problem, plan to develop model for predicting chornic kidney disease using LSTM, BiLSTM, BiGRU.	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement critically analyzed, the process model identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using th unified modelling language whic involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the proj will be handled by the doctors and i future updates in our project can b done based on detection of CKD.	PO4, PO7
C32SC4.3	The physical design includes webpage to predict whether the given input are CKD or NOT CKD.	PO5, PO6

Enhance Chronic Kidney Disease Detection Via Eurygasters Optimization And Ensemble Learning

ABSTRACT

Relative to the rest of the world, Chronic Kidney Disease remains one of the most important health challenges with a significant burden of morbidity and mortality. The disease presents a challenge; it alerts health systems because of its widespread impact on the health out comes of patients. However, CKD is mostly asymptomatic and the cause is usually diagnosed late. The paper proposed the new approach in the detection of early CKD, Eurygasters Optimization Algorithm with En semble Deep Learning, for the improvement of accuracy and reliability of the predictive models concerning CKD. The EOAEDL combines feature selection and the hyperparameter tuning process to boost detection ac curacy. It aims to select the relevant attributes from the medical dataset by carrying out the EOA, and the process of tuning hyperparameters is outsourced to the Shuffled Frog Leaping Algorithm. This approach is considered a hybrid of several deep learning models since it combines LSTM, BiGRU, and BiLSTM networks. These ensemble models are used to capture the critical temporal dependencies in medical data, which is critical for proper detection of CKD. In experiments on a benchmark dataset for CKD, the proposed EOAEDL algorithm has superior perfor mance in terms of accuracy, precision, and recall rates.

Keywords: Chronic Kidney Disease (CKD) prediction , Early-stage kidney diagnosis, Machine learning classification, Ensemble machine learning models, Feature extraction and selection.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Chronic Kidney Disease (CKD) is a progressive condition that affects millions of people worldwide and is associated with severe health complications, including kidney failure and cardiovascular diseases. Early detection and timely intervention are crucial for managing CKD and preventing its progression to end-stage renal disease [7].

However, one of the significant challenges in CKD diagnosis is that it remains asymptomatic in its early stages, making it difficult to detect without effective screening methodologies [11].

As the global prevalence of CKD continues to rise, there is a pressing need for advanced diagnostic tools that can facilitate early detection, allowing healthcare professionals to implement timely treatments and improve patient outcomes [6].

In recent years, artificial intelligence (AI) and machine learning (ML) have emerged as transformative technologies in healthcare, providing enhanced accuracy and efficiency in disease detection and diagnosis 7]. Deep learning (DL), a subset of ML, has demonstrated significant potential in analyzing complex medical data and identifying patterns that traditional methods might overlook [10].

This study aims to leverage DL techniques to develop a robust model for the early detection of CKD. Specifically, a deep neural network (DNN) is designed to analyze large-scale patient data, learning from intricate patterns to enhance predictive accuracy [17]. The proposed model aims to support clinicians by providing reliable diagnostic insights, ultimately enabling better decision-making and patient management [21].

Traditional diagnostic methods for CKD rely on biochemical tests, imaging, and clinical assessments, which may not always be efficient in early detection. These approaches are often time-consuming, prone to human error, and may not provide comprehensive insights into disease progression [2].

By integrating DL methodologies, particularly DNNs, healthcare professionals can improve predictive accuracy and reduce the reliance on manual interpretation of data. DNNs are particularly suited for medical applications because they can process large volumes of data while capturing non-linear relationships within datasets [5].

Unlike traditional ML models that rely heavily on manual feature selection, DNNs automatically extract features from raw medical data, allowing for a more nuanced and comprehensive analysis.

This capability is especially beneficial in CKD detection, where multiple biomarkers and risk factors must be considered simultaneously. By training the DNN on diverse datasets, the model can generalize well across different patient populations, increasing its effectiveness in real-world clinical settings [23].

Additionally, DNN models can be trained on electronic health records (EHRs), medical imaging, and laboratory results, allowing them to recognize patterns that may not be immediately evident to human practitioners. This ability to analyze multiple data sources simultaneously provides a more comprehensive and accurate diagnostic assessment [26].

One of the critical aspects of improving model performance in medical diagnostics is selecting the most relevant features. High-dimensional data can introduce noise, increase computational costs, and reduce model interpretability.

To address these challenges, this study employs Recursive Feature Elimination (RFE), a feature selection technique that systematically removes less relevant attributes while retaining those with the highest predictive value [6].

RFE plays a crucial role in refining the DNN model by ensuring that only the most informative features contribute to the predictive analysis. By eliminating redundant or less impactful variables, the model can achieve higher accuracy and efficiency in diagnosing CKD. Moreover, the reduction in dimensionality improves the model's interpretability, making it easier for healthcare professionals to understand the underlying factors influencing CKD predictions [19].

The combination of feature selection and deep learning enhances the robustness of the proposed framework, ensuring that it remains both effective and scalable for real-world applications. Additionally, incorporating feature selection techniques allows the model to prioritize essential biomarkers such as creatinine levels, glomerular filtration rate (GFR), and blood urea nitrogen (BUN), which are crucial indicators of kidney function [7].

In addition to employing deep learning techniques, this research explores the potential of integrating Internet of Things (IoT) and cloud-based technologies into CKD detection and management. IoT-enabled medical devices can continuously collect patient health data, including vital signs, biochemical parameters, and lifestyle factors [7] [8].

When combined with a cloud-based infrastructure, this real-time data can be processed and analyzed using the DL model, enabling continuous monitoring of patients' health status. The integration of IoT with cloud computing offers several benefits for CKD detection and management [10].

First, it enhances accessibility by enabling remote monitoring, allowing healthcare providers to track patients' conditions without the need for frequent hospital visits. Second, real-time data collection facilitates proactive interventions, as the system can detect early signs of CKD deterioration and alert physicians to take necessary action. Third, a cloud-based infrastructure allows seamless storage, retrieval, and sharing of medical data, fostering collaborative decision-making among healthcare professionals [22].

By leveraging IoT and cloud technologies, the proposed DL framework extends beyond traditional diagnostic approaches and embraces a more dynamic, patient-centered model of care. This shift not only improves early detection rates but also promotes personalized treatment plans, ultimately enhancing patient outcomes and quality of life [7].

The effectiveness of a deep learning model in CKD detection also depends on the quality and diversity of the dataset used for training and validation. Large-scale datasets with diverse patient demographics, medical histories, and laboratory results ensure that the model generalizes well to different populations [9].

To achieve this, data augmentation techniques can be employed to synthetically expand the dataset and improve model robustness. Moreover, balancing the dataset by addressing class imbalances where positive CKD cases may be significantly fewer than negative casescan prevent the model from being biased toward the majority class [22].

2. LITERATURE SURVEY

The literature survey provides a comprehensive overview of existing research and methodologies related to chronic kidney disease (CKD) detection and management, highlighting the evolution of diagnostic techniques and the role of machine learning (ML) and deep learning (DL) in enhancing predictive accuracy [7]. The increasing prevalence of CKD worldwide has prompted researchers to explore novel approaches that can facilitate early diagnosis and timely intervention, ultimately improving patient outcomes. Traditional CKD diagnosis has historically relied on conventional clinical methods, including blood tests and imaging techniques. However, these methods often fall short in detecting CKD at its early stages, as the disease can progress silently without significant symptoms [10].

Many patients remain undiagnosed until they reach an advanced stage, making it challenging to implement preventive measures. Consequently, the need for more sophisticated and data-driven approaches to CKD diagnosis has become evident, leading to the integration of ML and DL techniques into healthcare systems.Several researchers have explored various ML algorithms for CKD prediction. ML-based approaches have proven to be effective in analyzing large datasets and identifying hidden patterns that might indicate the presence of CKD [11].

Arulanthu and Perumal (2021) developed an intelligent chronic kidney disease diagnosis system utilizing cloud-centric optimal feature subset selection combined with novel data classification models. Their work demonstrated the effectiveness of ML in enhancing diagnostic accuracy, emphasizing the importance of feature selection in improving model performance [9].

Similarly, Lambert et al. (2020) investigated the impact of different ML classification algorithms on CKD, revealing that ensemble methods often outperform single classifiers in terms of predictive accuracy. The effectiveness of ML models in CKD diagnosis is largely influenced by the quality of feature selection and the ability of algorithms to distinguish between relevant and redundant variables [22].

Studies suggest that incorporating feature selection techniques such as Recursive Feature Elimination (RFE) can significantly enhance model interpretability and reduce computational complexity, making ML-based diagnostic tools more efficient and accessible in clinical settings [22].

In the realm of deep learning, significant advancements have been made in the application of neural networks for CKD detection. Bhaskar and Manikandan (2019) proposed a deep-learning-based system for automated sensing of CKD, showcasing the potential of convolutional neural networks (CNNs) in analyzing medical images and patient data. Their findings indicated that deep learning models could achieve higher accuracy compared to traditional ML methods, particularly in complex datasets [10].

The ability of deep learning models to process large amounts of structured and unstructured data has made them an attractive choice for CKD diagnosis. CNNs, recurrent neural networks (RNNs), and hybrid deep learning architectures have been employed in several studies to improve predictive accuracy and automate the diagnostic process [8].

Deep learning models, unlike traditional ML algorithms, can automatically extract relevant features from raw medical data, reducing the dependency on manual feature selection. This characteristic makes them particularly valuable in scenarios where CKD diagnosis relies on analyzing intricate patterns in patient records, laboratory results, and imaging data [1].

Moreover, the literature highlights the importance of hybrid models that combine multiple algorithms to enhance predictive capabilities. Hybrid models leverage the strengths of different approaches, compensating for the limitations of individual techniques. Ramesh and Lakshmanna (2024) introduced a novel framework for early detection and prevention of coronary heart disease using a hybrid deep learning model and neural fuzzy inference system [7]. This approach underscores the potential of integrating various techniques to improve diagnostic accuracy and provide comprehensive patient assessments. Hybrid models have been widely adopted in CKD research to optimize predictive performance.

For example, combining decision trees with deep learning networks has been shown to improve classification accuracy, while integrating genetic algorithms with ML models has facilitated better feature selection and enhanced robustness.By employing hybrid approaches, researchers can create more reliable and efficient diagnostic models that cater to diverse patient populations and medical environments [22].

The survey also addresses the role of IoT and cloud computing in CKD management. The integration of these technologies allows for real-time data collection and monitoring, facilitating proactive healthcare management. For instance, Noor et al. (2019) developed an IoT-based mHealth platform for CKD patients, enabling continuous monitoring and timely interventions [23]. This approach not only enhances patient engagement but also supports healthcare providers in making informed decisions based on real-time data. IoT-enabled devices, such as wearable sensors and smart medical instruments, can track vital signs, dietary habits, and medication adherence, providing clinicians with continuous insights into a patient's health status.

When combined with cloud computing, these data streams can be processed in real-time, allowing healthcare professionals to detect early warning signs of CKD progression and intervene before the condition worsens. Cloud-based platforms also facilitate seamless data sharing between healthcare providers, improving collaboration and ensuring that patients receive personalized and timely care [6].

Despite the promising advancements in ML and DL for CKD detection, challenges remain. Many existing models are limited by their dependency on sequential techniques, which may hinder scalability to larger datasets. Additionally, the need for extensive labeled data for training deep learning models poses a significant barrier to widespread implementation. Data scarcity and class imbalance issues further complicate model training, leading to potential biases in predictive performance. Future research should focus on addressing these limitations by exploring techniques that enhance computational efficiency and model scalability for real-time applications [5].

Data augmentation strategies play a crucial role in addressing the limitations associated with insufficient and imbalanced medical datasets. Generative adversarial networks (GANs) have emerged as a powerful tool for generating synthetic medical data that closely resembles real-world patient records. By training a generator and a discriminator network simultaneously, GANs can create high-quality synthetic samples that enhance the diversity of training data. This approach helps mitigate the challenges posed by limited labeled data and improves the robustness of machine learning models used for CKD detection [2].

In addition to GANs, other synthetic data generation techniques, such as variational autoencoders (VAEs) and data interpolation methods, have been explored to augment training datasets. These techniques facilitate the creation of diverse patient profiles, enabling deep learning models to generalize better across different demographics. By introducing slight variations in training data, these methods help reduce overfitting, ensuring that models perform reliably when exposed to real-world medical cases. Furthermore, synthetic data generation allows researchers to simulate rare CKD cases that might be underrepresented in actual datasets, enhancing the model's capability to detect uncommon manifestations of the disease.Federated learning has emerged as a promising approach to overcoming data privacy concerns while enabling collaborative model training across multiple healthcare institutions [4].

3. SYSTEM ANALYSIS

The proposed system focuses on the early detection of Chronic Kidney Disease (CKD) using an ensemble deep learning approach combined with the Eurygasters Optimization Algorithm (EOA). Traditional diagnostic methods for CKD are often inefficient, leading to late-stage detection and poor patient outcomes.

This system aims to improve accuracy, precision, and recall by leveraging advanced machine learning and deep learning techniques. The primary objective is to enhance feature selection and classification to develop a scalable and efficient CKD detection framework.

The system processes a clinical dataset comprising 400 patient records, with 250 CKD cases and 150 non-CKD cases. It includes demographic attributes such as age, blood pressure, and diabetes status, along with medical test results like serum creatinine, hemoglobin, blood urea, and red/white blood cell counts.

The preprocessing stage involves min-max normalization to scale numerical data between 0 and 1, handling missing values, and selecting relevant features using EOA. This feature selection process helps reduce dimensionality, improve interpretability, and minimize overfitting.

To achieve accurate classification, the system integrates three deep learning models: Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU). LSTM captures long-term dependencies in sequential data, while BiLSTM enhances contextual understanding by reading data in both forward and backward directions.

BiGRU, an efficient alternative to LSTM, captures sequential dependencies with fewer parameters. The ensemble learning approach combines these models, improving classification accuracy by averaging their outputs.

Furthermore, hyperparameter tuning is performed using the Shuffled Frog Leaping Algorithm (SFLA), which optimizes parameters such as learning rate, dropout rate, and batch size. This optimization ensures better model convergence and enhances classification performance.

Once trained, the system classifies patients as CKD-positive or non-CKD based on the ensemble model's predictions. The proposed system achieves high accuracy, with performance metrics reaching up to 98.75% accuracy, along with improved precision, recall, and F1-score compared to conventional machine learning models such as SVM, ANN, and k-NN.

The workflow of the system follows a structured process: clinical data is collected, preprocessed through normalization and feature selection, and used to train the deep learning models. Hyperparameter tuning is applied to optimize model parameters before making CKD predictions.

The system's comparative analysis demonstrates that it significantly outperforms traditional approaches, providing higher sensitivity and specificity in CKD diagnosis. The use of bio-inspired optimization techniques, such as EOA and SFLA, further enhances feature selection and classification accuracy.

For future improvements, the system can be integrated with real-time healthcare monitoring systems, such as wearable devices, for continuous patient monitoring. Additionally, its methodology can be expanded to diagnose other chronic diseases using similar optimization and deep learning strategies.

Implementing the model in cloud-based medical applications would allow for remote diagnosis, making it accessible for widespread clinical use. Overall, the proposed CKD detection framework presents a robust, scalable, and highly accurate solution for early disease detection and improved patient outcomes.

3.1 EXISTING SYSTEM

The conventional method for diagnosing Chronic Kidney Disease (CKD) is based on clinical examinations, laboratory tests, and physician assessments. Doctors analyze key biomarkers such as serum creatinine levels, glomerular filtration rate (GFR), urine protein levels, and blood pressure to determine kidney function [7]. These tests follow standard medical thresholds to classify patients as CKD-positive or non-CKD. While these methods are widely used and form the backbone of clinical diagnostics, they have several limitations, especially in detecting CKD at its early stages.

Since CKD often progresses silently, many patients remain undiagnosed until the disease has significantly advanced. At this stage, kidney damage may be irreversible, leading to the need for long-term treatments such as dialysis or kidney transplantation [10].

This delay in diagnosis is a major concern in healthcare because early detection plays a crucial role in slowing disease progression and improving patient outcomes. Unfortunately, traditional diagnostic methods heavily rely on manual interpretation by physicians, making them prone to human error, variations in medical expertise, and inconsistencies in diagnosis [1].

To address these challenges, machine learning (ML) models have been introduced as an alternative approach for CKD detection. ML models such as Logistic Regression (LR), Support Vector Machines (SVM), Naïve Bayes (NB), and k-Nearest Neighbors (k-NN) analyze patient data and identify CKD cases based on patterns found in clinical variables. These models process large amounts of structured data, including demographics, blood test results, and urine test parameters, to make predictions about CKD risk [8].

Unlike human diagnosis, ML models can discover hidden patterns and relationships that may not be immediately apparent to medical professionals. By training on labeled datasets, these algorithms attempt to predict CKD cases with improved accuracy. However, traditional ML models have limitations, particularly when dealing with complex medical data [2].

Many of these models struggle with nonlinear feature interactions, meaning they may not effectively capture how different patient attributes contribute to CKD progression. Additionally, some ML models, such as Decision Trees, may perform well on training data but fail to generalize to new, unseen patient cases, leading to overfitting and unreliable predictions [6].

To further improve CKD classification, researchers have developed hybrid optimization models that integrate feature selection techniques. Feature selection is a critical step in machine learning because it helps identify the most relevant medical variables while eliminating redundant or unimportant features. One commonly used optimization method is Particle Swarm Optimization (PSO), which mimics the behavior of a swarm of particles searching for the best solution in a given problem space [7].

Another popular approach is Genetic Algorithms (GA), which uses the principles of natural selection and evolution to optimize feature selection. These optimization techniques enhance model accuracy while reducing computational complexity, making them attractive for medical applications. However, these techniques also come with drawbacks. PSO and GA often experience slow convergence rates, meaning they require a large number of iterations to find the best set of features. Additionally, these optimization methods are static, meaning they do not dynamically adapt to new patient data, limiting their effectiveness in real-world clinical applications where CKD progression patterns may change over time [22].

The need for an improved CKD detection system is evident, as existing methods have major drawbacks in early diagnosis, accuracy, feature selection, and computational efficiency. A more advanced approach would involve ensemble deep learning, feature selection optimization, and dynamic hyperparameter tuning to improve CKD classification performance [4].

By leveraging multiple deep learning models in an ensemble framework, such a system can enhance accuracy, increase robustness, and provide better generalization to real-world patient data.

Additionally, integrating an intelligent optimization algorithm for hyperparameter tuning can significantly reduce training time and computational cost, making the system more practical for large-scale medical applications. A well-designed CKD detection model has the potential to assist doctors by providing faster and more accurate predictions, ultimately leading to better patient care, early treatment interventions, and improved quality of life for individuals at risk of kidney disease [22].

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

Despite significant advancements in CKD detection methods, both traditional clinical approaches and machine learning models still face multiple limitations. One of the most pressing issues is delayed diagnosis, which occurs because CKD often progresses silently, with no noticeable symptoms in its early stages [7].

Traditional methods primarily rely on clinical tests such as serum creatinine levels, glomerular filtration rate (GFR), urine protein levels, and blood pressure monitoring, but these tests are usually conducted only when a patient starts exhibiting symptoms. By the time CKD is detected through these methods, significant and often irreversible kidney damage may have already occurred. This limits treatment options, making early intervention difficult and leading to a higher risk of progression to end-stage renal disease (ESRD), which requires dialysis or kidney transplantation [7] [2].

Moreover, the manual nature of traditional diagnostic procedures makes them time-consuming, requiring multiple visits to healthcare facilities, numerous laboratory tests, and expert interpretation by nephrologists. The dependency on human expertise introduces the possibility of inconsistencies, misinterpretations, and diagnostic errors, which can result in misdiagnosis or delayed treatments. Furthermore, these manual approaches may vary between healthcare

Despite advances in ML and deep learning, most existing CKD detection systems still rely on single classifiers, which limits their generalization capabilities. A single classifier, whether it is a decision tree, neural network, or SVM, may not be able to fully capture the complexity and variability of CKD data, resulting in lower prediction accuracy. Ensemble learning, which combines multiple models to improve classification robustness, has been shown to be far more effective than using a single model. However, many CKD detection systems fail to implement ensemble learning, which prevents them from achieving higher accuracy and adaptability. Furthermore, another major limitation is the lack of effective hyperparameter tuning techniques. Hyperparameters, such as learning rate, batch size, and activation functions, play a crucial role in model performance [7].

Most existing CKD detection models still rely on manual or trial-and-error hyperparameter tuning, which is both inefficient and time-consuming. Without proper tuning, models may fail to achieve optimal performance, leading to poor accuracy, increased misclassification rates, and limited clinical usability [20].

These limitations highlight the urgent need for an optimized, ensemble-based CKD detection system that integrates deep learning, feature selection optimization, and dynamic hyperparameter tuning. By leveraging multiple deep learning models within an ensemble framework, such a system can significantly improve accuracy, increase robustness, and generalize better to real-world patient data [25].

Additionally, incorporating intelligent hyperparameter tuning algorithms can reduce training time and computational costs, making the system more practical for large-scale deployment in healthcare facilities. A well-designed CKD detection system has the potential to assist doctors by providing faster and more accurate predictions, ultimately leading to better patient outcomes, early treatment interventions, and improved quality of life for individuals at risk of CKD [19].

3.2 PROPOSED SYSTEM

The proposed system aims to overcome the limitations of traditional clinical diagnosis, standalone machine learning models, and deep learning approaches in detecting Chronic Kidney Disease (CKD). By integrating ensemble deep learning, feature selection optimization, and hyperparameter tuning, this system provides a more accurate, efficient, and scalable solution for early CKD detection. Unlike conventional diagnostic approaches that rely on manual medical assessments, which are prone to human error and delayed diagnosis, this system leverages automated data processing and artificial intelligence to improve detection accuracy.

The core of the proposed system lies in its ability to analyze large patient datasets, select the most relevant features, and use a combination of deep learning models to improve classification performance. Through this approach, it enhances CKD detection in its early stages, allowing for timely intervention and reducing the risk of disease progression.

The first key component of the proposed system is data preprocessing and feature selection. The dataset consists of 400 patient records, with 250 CKD cases and 150 non-CKD cases, covering crucial clinical parameters such as age, blood pressure, serum creatinine levels, hemoglobin concentration, and urine analysis results. To ensure that the data is clean and suitable for machine learning models, missing values are either imputed or removed, and all numerical features undergo Min-Max normalization, scaling them within a fixed range of 0 to 1.

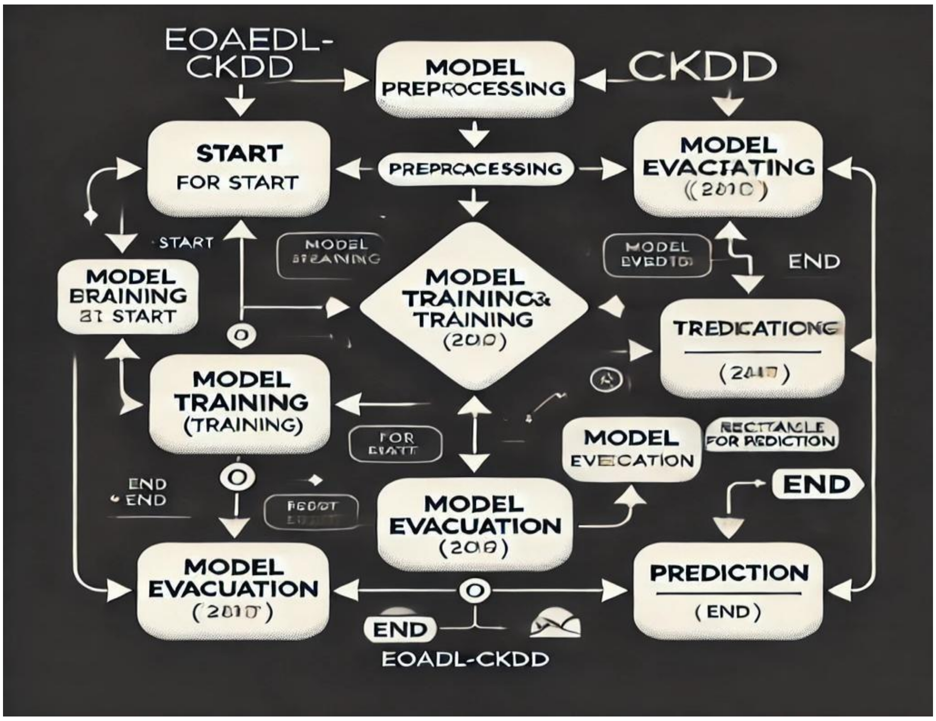


Fig. 1: flow chat for proposed model

Feature selection is a crucial step in improving the system's efficiency and accuracy, as irrelevant or redundant features can introduce noise and reduce model performance. To address this, the Eurygasters Optimization Algorithm (EOA) is implemented. Inspired by the foraging behavior of Eurygaster beetles, this algorithm efficiently selects the most informative medical attributes while eliminating unnecessary variables. By optimizing the input feature set, the computational complexity is reduced, leading to a more robust and interpretable CKD prediction model.

Once the dataset has been processed and optimized, the system employs an ensemble of deep learning models for classification. The primary models used include Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU). These deep learning architectures are specifically chosen for their ability to capture temporal dependencies in sequential medical data, making them ideal for diseases like CKD, where progression trends over time are critical for accurate diagnosis.

LSTM is widely known for its ability to retain long-term dependencies, ensuring that historical patient data plays a role in the decision-making process. BiLSTM enhances this capability by processing input sequences in both forward and backward directions, allowing the model to better understand contextual relationships in patient records.

Meanwhile, BiGRU is a computationally lighter variant that achieves similar performance with fewer parameters, making the system more efficient without compromising accuracy. Instead of relying on a single deep learning model, the proposed system combines the outputs of LSTM, BiLSTM, and BiGRU in an ensemble framework, leveraging their individual strengths to achieve higher classification accuracy and reduced error rates.

To further enhance performance, the proposed system implements hyperparameter optimization using the Shuffled Frog Leaping Algorithm (SFLA). Hyperparameters, such as learning rate, batch size, dropout rate, and activation functions, significantly impact a deep learning model's ability to learn effectively and generalize to unseen data. Rather than relying on traditional manual tuning or trial-and-error approaches, SFLA is used to automatically search for the best hyperparameter configurations, ensuring that the system reaches optimal performance with minimal human intervention.

SFLA is inspired by the natural behavior of frogs searching for food, where each frog represents a potential solution, and the best-performing solutions are continuously refined through iterative adjustments.

In conclusion, the proposed ensemble deep learning system for CKD detection successfully addresses the challenges of existing approaches by integrating feature selection optimization (EOA), ensemble deep learning (LSTM, BiLSTM, BiGRU), and intelligent hyperparameter tuning (SFLA).

3.3 FEASIBILITY STUDY

The feasibility study examines whether the proposed ensemble deep learning system for CKD detection is practical, effective, and suitable for real-world implementation. Since early CKD detection plays a crucial role in preventing disease progression and reducing healthcare costs, this study evaluates the technical, economic, operational, scalability, and legal feasibility of the proposed system. The goal is to determine if the system can be seamlessly integrated into clinical workflows while maintaining high accuracy, efficiency, and affordability.

Technical Feasibility

The proposed system is technically feasible as it leverages state-of-the-art deep learning models combined with bio-inspired feature selection and hyperparameter optimization techniques. It integrates Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU) to classify CKD cases with high precision.

These deep learning architectures are well-suited for analyzing sequential medical data, making them particularly effective for CKD prediction, where patient history and disease progression over time play a significant role. Unlike traditional models that struggle with long-term dependencies in patient records, the proposed deep learning ensemble can efficiently extract meaningful patterns, leading to more accurate and reliable diagnoses.

Additionally, the system includes an Eurygasters Optimization Algorithm (EOA) for feature selection, which automatically selects the most relevant medical features while eliminating redundant or less impactful attributes.This not only improves classification accuracy but also reduces computational complexity, ensuring that the system remains efficient even when processing large-scale patient datasets.

Furthermore, the implementation of the Shuffled Frog Leaping Algorithm (SFLA) for hyperparameter tuning ensures that the model is trained under optimal conditions, reducing overfitting and enhancing generalization to new patient data.From a technological standpoint, all the required machine learning frameworks, cloud computing resources, and data processing tools are readily available, making the system highly implementable in real-world healthcare environments.

The cloud-based compatibility of the proposed model ensures that it can be integrated into existing hospital infrastructures, allowing real-time access to CKD predictions. The system is also designed to work efficiently on standard computing hardware, meaning that hospitals and clinics do not require expensive high-performance computing systems to implement it.

Economic Feasibility

The economic feasibility of the proposed system is a major advantage, as it significantly reduces the financial burden associated with CKD diagnosis and long-term management. Traditional CKD detection requires multiple laboratory tests, specialist consultations, and manual data interpretation, leading to high diagnostic and operational costs. By automating the diagnostic process using AI-driven models, the proposed system eliminates the need for repeated testing, reducing costs for both patients and healthcare providers.

The implementation of feature selection (EOA) and hyperparameter tuning (SFLA) plays a crucial role in minimizing computational expenses. By selecting only the most critical medical attributes, the system processes less data while maintaining high accuracy, reducing the amount of computational power required.Additionally, the proposed system is scalable, meaning that once trained, the AI model can process a large number of patient records simultaneously without incurring additional costs. This is particularly beneficial for large hospitals and national health programs, where screening thousands of patients for CKD in an affordable manner is a necessity.

Although there may be an initial investment required for AI training, model deployment, and cloud-based infrastructure, the long-term cost savings outweigh these upfront expenses. Over time, the system can prevent late-stage CKD cases, reducing the number of patients who require expensive dialysis treatments or kidney transplants.

In developing regions, where specialist nephrologists and advanced diagnostic tools may be scarce, this AI-driven system offers a cost-effective solution for mass screening programs, improving CKD diagnosis while keeping healthcare expenditures manageable.

Operational Feasibility

Operational feasibility is a critical factor in determining whether the system can be efficiently integrated into existing medical workflows. The proposed system is designed to be user-friendly, requiring minimal manual intervention from healthcare professionals. It can be integrated into hospital information systems (HIS) and electronic health records (EHRs), ensuring that doctors and medical staff can easily access AI-generated CKD risk assessments.

Unlike traditional ML models that require extensive manual feature engineering and preprocessing, the proposed system automates these tasks, allowing for faster and more reliable CKD classification. The ensemble deep learning approach ensures that predictions are highly accurate, reducing the likelihood of false positives and false negatives, which are common in conventional CKD screening techniques. Additionally, the system is built to be adaptive, meaning it can be continuously updated with new patient data, improving its predictive capabilities over time.

Healthcare professionals require minimal training to operate the system, as it is designed to provide clear, interpretable results. AI-based CKD detection acts as a decision-support tool, assisting doctors rather than replacing them.

It offers insights into disease progression trends, allowing nephrologists and general practitioners to make informed treatment decisions while reducing diagnostic workload. The system is also optimized for real-time CKD detection, meaning that as soon as a patient’s lab results are entered, the AI model can generate a risk assessment within seconds, expediting the decision-making process.

Scalability Feasibility

One of the most significant advantages of the proposed system is its scalability. Since it is built on deep learning frameworks compatible with cloud computing, it can handle large datasets and multiple patient records simultaneously. This makes it suitable for small clinics, large hospitals, and even national health screening programs.

The feature selection process (EOA) ensures that the system remains computationally efficient, making it applicable to both high-resource and low-resource healthcare settings. In rural areas or regions with limited access to specialist nephrologists, this AI system can be deployed as a cloud-based diagnostic service, allowing doctors to remotely assess CKD risk.

Furthermore, as the Internet of Things (IoT) and wearable health monitoring devices become more prevalent, this system can be adapted to integrate real-time patient monitoring, enabling continuous CKD risk assessment based on real-world biometrics [10].

4. SYSTEM REQUIREMENT

The proposed AI-driven CKD detection system requires a well-defined set of system requirements to ensure efficient performance, accuracy, and scalability. These requirements include hardware, software, data, functional, and security specifications necessary for optimal model execution, real-time prediction, and seamless integration into healthcare environments.

Given the complexity of handling large-scale medical data and deep learning models, the system must be equipped with the necessary computing resources while remaining cost-effective and adaptable for different healthcare settings.

4.1 SOFTWARE REQUIREMENTS

The proposed AI-driven CKD detection system requires a robust and scalable software framework to support deep learning, data processing, and integration with healthcare systems. The operating system must be stable and compatible with machine learning frameworks, supporting both local and cloud-based deployments.

For integration with hospitals, the system will use Flask or FastAPI to develop RESTful APIs, allowing seamless connection with existing EHR systems and hospital networks.

Windows 10/11 is suitable for local execution and model testing, while Ubuntu 20.04+ is recommended for cloud-based high-performance computing environments. Additionally, cloud platforms such as Google Cloud, AWS, or Microsoft Azure can be used for large-scale AI training and inference.

The system is primarily developed in Python 3.x, which is widely used in artificial intelligence and healthcare applications. Python provides access to essential deep learning frameworks like TensorFlow and PyTorch, allowing the development of LSTM, BiLSTM, and BiGRU models for CKD classification.

Additionally, Keras is used for high-level neural network implementation, while Scikit-learn is utilized for feature selection using Eurygasters Optimization Algorithm (EOA) and hyperparameter tuning with Shuffled Frog Leaping Algorithm (SFLA). Data manipulation is handled through NumPy and Pandas, while Matplotlib and Seaborn are used for visualizing model performance and CKD prediction insights.

4.2 REQUIREMENT ANALYSIS

The requirement analysis of the proposed AI-driven CKD detection system ensures that it meets the technical, functional, and security standards necessary for real-world deployment in hospitals, clinics, and research institutions. The system is designed to automate CKD detection by leveraging deep learning models and providing real-time risk assessment. It must be capable of processing patient data, predicting CKD risk, and integrating with hospital management systems. The system accepts structured input such as patient demographics, blood test results, and urine analysis data and processes it using feature selection algorithms (EOA) and deep learning models (LSTM, BiLSTM, BiGRU) to classify patients as CKD-positive or non-CKD. Furthermore, the system must support API integration with electronic health records (EHRs), ensuring a seamless flow of medical information for real-time decision-making in clinical settings [7].

In addition to its core functionalities, the system must meet non-functional requirements to ensure efficiency, scalability, and usability. The system should generate CKD predictions within 2 seconds, allowing doctors to make quick and informed clinical decisions. It must also be scalable, capable of handling large volumes of patient records, making it suitable for individual hospitals, national health screening programs, and cloud-based medical applications. A user-friendly graphical interface should be designed to simplify navigation for medical professionals, ensuring accessibility without requiring advanced technical expertise. The system should also support multi-language interfaces to cater to diverse healthcare settings worldwide.

The hardware requirements of the system depend on its mode of deployment. For local execution and inference, the system is designed to run on a Lenovo IdeaPad Slim 3 (15.7") with 8GB RAM and 256GB SSD, making it suitable for running pre-trained AI models and performing small-scale testing. However, for large-scale model training and processing, cloud-based GPU servers such as Google TPUs or NVIDIA Tesla A100 GPUs are recommended. These cloud-based solutions allow hospitals and research institutions to train models faster and process large patient datasets efficiently. The system also requires high-speed internet connectivity to ensure real-time cloud access and seamless data retrieval from hospital databases.

On the software side, the system requires a robust stack of programming tools, deep learning frameworks, and database management systems. The operating system should be Windows 10/11 for local execution and Ubuntu 20.04+ for cloud-based AI deployments. The core development is done using Python 3.x, which supports essential machine learning frameworks detection system. The system integrates deep learning models (LSTM, BiLSTM, BiGRU), feature selection (EOA), and hyperparameter tuning (SFLA) to ensure high accuracy and reliability in CKD risk assessment. The hardware and software configurations allow for local execution on Lenovo IdeaPad Slim 3, while cloud-based deployments ensure scalability for large hospitals and national screening programs. The integration of secure database management and encryption mechanisms ensures patient confidentiality and compliance with healthcare laws. Overall, this AI-powered CKD detection system is well-equipped for real-world adoption, providing fast, reliable, and cost-effective early-stage CKD diagnosis, ultimately improving patient outcomes and reducing the burden on healthcare professionals.

4.3 HARDWARE REQUIREMENTS

The proposed AI-driven CKD detection system requires a well-defined hardware infrastructure to support data processing, model training, and real-time inference. The hardware must be capable of handling deep learning models, feature selection algorithms (EOA), and hyperparameter tuning (SFLA) while ensuring efficient storage, fast processing, and seamless integration with hospital networks.

- Processor: Intel Core i5 / AMD Ryzen 5 or higher
- RAM: 8GB DDR4 (sufficient for inference, but cloud extensive model training)4.3
- Storage: 256GB SSD (fast read/write speeds for model access
- GPU: Integrated Graphics, Google colab pro (but deep learning training requires cloud-based GPUs)
- Internet Connectivity: Required for real-time model updates, API integration, and cloud access

4.4 SOFTWARE

The AI-powered CKD detection system is built using a combination of programming languages, deep learning frameworks, database management systems, and API development tools. Each software component plays a critical role in enabling data processing, feature selection, deep learning model execution, security, and real-time deployment. Below is an overview of the key software used and how each contributes to the functioning of the system.

Programming Language – Python 3.x

Python 3.x serves as the core programming language due to its extensive library support, scalability, and compatibility with AI models. It enables smooth interaction with machine learning frameworks and hospital management systems, ensuring that CKD risk assessments can be performed accurately and efficiently.

Deep Learning Frameworks – TensorFlow and PyTorch

For deep learning, the system utilizes TensorFlow 2.x and PyTorch, which are essential for building, training, and deploying AI models. The deep learning models, including LSTM, BiLSTM, and BiGRU, are trained to analyze sequential medical data and detect patterns associated with CKD progression. These frameworks support GPU acceleration, significantly improving computational efficiency, especially during model training and inference. TensorBoard, an additional tool within TensorFlow, helps in visualizing model performance, monitoring loss functions, and tracking accuracy improvements over time.

Data Processing and Manipulation – NumPy and Pandas

Feature selection and optimization are key components of the system, handled by Scikit-learn. This library is used for preprocessing patient data, selecting the most relevant CKD risk factors using the Eurygasters Optimization Algorithm (EOA), and fine-tuning model hyperparameters through the Shuffled Frog Leaping Algorithm (SFLA). NumPy and Pandas further assist in handling large medical datasets, ensuring that all patient records are cleaned, structured, and formatted properly before being analyzed by the AI model.

API Development – Flask and FastAPI

Additionally, Flask and FastAPI facilitate the development of RESTful APIs, enabling hospitals and research centers to access real-time CKD predictions from the AI model. FastAPI is particularly useful due to its high-speed API response times, ensuring that risk assessments are available almost instantly when requested by medical professionals.

4.5 SOFTWARE DESCRIPTION

The AI-driven CKD detection system is a sophisticated software solution designed to assist healthcare professionals, researchers, and hospitals in accurately identifying Chronic Kidney Disease (CKD) at an early stage. The system utilizes machine learning, deep learning, feature selection, and optimization techniques to analyze structured medical data, providing real-time predictions and clinical insights. Built using Python and deep learning frameworks such as TensorFlow and PyTorch, the software is capable of handling large patient datasets, ensuring high accuracy, scalability, and seamless integration with existing hospital management systems.

The core functionality of the software includes data preprocessing, feature selection, model execution, and real-time inference. The system first ingests patient records, either from electronic health records (EHRs), hospital databases, or manual data entry, and processes them to ensure data consistency and integrity. This involves cleaning missing values, normalizing data using Min-Max Scaling, and selecting the most relevant CKD risk factors using the Eurygasters Optimization Algorithm (EOA). By filtering out unnecessary attributes, the system improves classification accuracy and reduces computational overhead.

At the heart of the system lies its ensemble deep learning model, which consists of Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU). These models are specifically designed to handle sequential medical data, allowing the system to recognize patterns and trends in CKD progression over time. Instead of relying on a single model, the system combines multiple deep learning architectures, leveraging their strengths to produce highly accurate and reliable CKD risk predictions. The final output is a probability score indicating the likelihood of CKD, along with feature importance rankings to help doctors understand the key risk factors influencing the prediction.

The user interface and accessibility of the software have been designed to support seamless integration with hospital systems. Using Flask or FastAPI, the software is deployed as a RESTful API, enabling real-time interactions between the AI model and clinical applications. Doctors, nurses, and medical staff can access AI-generated CKD risk assessments through web applications, mobile apps, or hospital dashboards, making it convenient to incorporate the software into routine clinical workflows. Additionally, data visualization tools such as Matplotlib and Seaborn provide clear, graphical representations of CKD risk trends, patient history insights, and model performance metrics, ensuring interpretability and transparency for medical professionals.

Security and compliance are critical aspects of the software, ensuring that sensitive patient data is protected while adhering to global healthcare regulations such as HIPAA (U.S.), GDPR (Europe), and local data protection laws. The software encrypts all medical records using AES-256 encryption, ensuring that patient data remains confidential and secure.

5. SYSTEM DESIGN

The EOAEDL-CKDD system is designed as an AI-powered framework for the accurate detection of chronic kidney disease (CKD). The system architecture is modular and comprises key components, including data preprocessing, feature selection, model training, hyperparameter tuning, and real-time prediction. The data acquisition module collects patient demographic details, laboratory test results, and clinical parameters from electronic health records (EHRs) and public medical databases. Since medical data often contains missing values and inconsistencies, preprocessing techniques such as missing value imputation, outlier detection, and Min-Max normalization are applied to ensure data quality. These preprocessing steps help standardize the dataset and improve the model's ability to learn meaningful patterns for CKD prediction.

To improve model performance and computational efficiency, the system incorporates a feature selection module using the Eurygasters Optimization Algorithm (EOA). This bio-inspired technique identifies the most significant clinical indicators associated with CKD while eliminating redundant feature [7]. By reducing the dimensionality of the dataset, EOA ensures that only the most relevant attributes, such as serum creatinine levels, blood pressure, and urine albumin concentration, are used for training, improving both accuracy and interpretability. The model training module leverages deep learning architectures, including Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM), to analyze sequential medical data. These models effectively capture temporal dependencies in patient health records, making them well-suited for detecting progressive diseases like CKD.

To further optimize model performance, the system integrates hyperparameter tuning using the Shuffled Frog Leaping Algorithm (SFLA). This nature-inspired optimization algorithm fine-tunes key hyperparameters, such as learning rate, batch size, and dropout rate, ensuring that the deep learning models converge optimally. Once training is complete, the evaluation module assesses model effectiveness using metrics like accuracy, precision, recall, F1-score, and AUC-ROC [7]. A high recall score ensures that CKD cases are identified correctly, minimizing false negatives. Additionally, cross-validation is conducted across multiple datasets to verify that the model generalizes well to different patient populations, improving its robustness for real-world applications.

The deployment module ensures that the system is accessible and scalable for healthcare applications. The AI-powered CKD detection model is integrated into a cloud-based web platform, allowing healthcare professionals to input patient data and receive real-time diagnostic insights. Furthermore, IoT-enabled wearable devices can continuously monitor vital health parameters such as blood pressure and kidney function markers, feeding real-time data into the model for continuous CKD risk assessment. By combining deep learning, feature optimization, and IoT integration, the EOAEDL-CKDD system offers a highly accurate and efficient solution for early CKD detection, significantly improving clinical decision-making and patient outcomes.

5.1 SYSTEM ARCHITECTURE

The EOAEDL-CKDD system architecture is structured to efficiently process patient health data for chronic kidney disease (CKD) detection using machine learning, deep learning, and optimization techniques. It begins with data acquisition and preprocessing, where medical records are cleaned using missing value imputation and normalization. The Eurygasters Optimization Algorithm (EOA) selects the most relevant features, reducing complexity and enhancing interpretability **[7]**. LSTM and BiLSTM models capture temporal dependencies in patient health records, while hyperparameter tuning with the Shuffled Frog Leaping Algorithm (SFLA) optimizes performance. The model is evaluated using accuracy, precision, recall, F1-score, and AUC-ROC for reliability. Finally, the system integrates with cloud-based platforms and IoT devices, enabling real-time CKD risk assessment and clinical decision support.

5.1.1 DATASET DESCRIPTION

The dataset utilized in this project is critical for the development and validation of the deep learning-based system for chronic kidney disease (CKD) detection. It comprises a comprehensive collection of patient medical records, laboratory test results, and demographic information, sourced from reputable healthcare institutions and publicly available medical databases.

The dataset includes a diverse range of features that are essential for accurately predicting CKD, such as age, gender, blood pressure, specific laboratory values (e.g., serum creatinine, blood urea nitrogen, and glomerular filtration rate), and medical history, including the presence of diabetes and hypertension.

The dataset is structured to include both categorical and continuous variables, allowing for a robust analysis of the factors contributing to CKD. Categorical variables, such as gender and diabetes status, provide essential demographic context, while continuous variables, such as serum creatinine levels, offer quantitative insights into kidney function. The dataset is balanced to ensure that both CKD-positive and CKD-negative cases are adequately represented, which is crucial for training machine learning models effectively [7] [19].

Data preprocessing steps are applied to enhance the quality of the dataset, including normalization to standardize the range of continuous variables, handling missing values through imputation techniques, and removing outliers that may skew the results. The dataset is then split into training, validation, and test sets to facilitate model training and evaluation.

This structured approach ensures that the models developed in this project are trained on high-quality data, ultimately leading to improved accuracy and reliability in predicting CKD. The comprehensive nature of the dataset, combined with rigorous preprocessing, positions it as a valuable resource for advancing CKD detection methodologies.

5.1.2 PREPROCESSING

Data preprocessing is a crucial step in the development of the chronic kidney disease (CKD) detection system, as it ensures the quality and reliability of the dataset used for training machine learning and deep learning models. The preprocessing phase begins with data cleaning, which involves identifying and addressing missing values within the dataset. Missing data can arise from various sources, such as incomplete medical records or errors during data collection. To handle this, imputation techniques are employed, such as mean or median imputation for continuous variables and mode imputation for categorical variables. This approach helps maintain the integrity of the dataset while minimizing the loss of valuable information.

Following the handling of missing values, the next step is to detect and remove outliers that may skew the analysis. Outliers can significantly impact model performance, leading to inaccurate predictions. Statistical methods, such as the Z-score method or the Interquartile Range (IQR) method, are utilized to identify these anomalies, which are then either removed or adjusted based on the context of the data.

Normalization is another critical preprocessing step, particularly for continuous variables. This process involves scaling the data to a standard range, typically between 0 and 1, using techniques such as min-max normalization. Normalization ensures that all features contribute equally to the model training process, preventing any single feature from disproportionately influencing the results due to its scale.

Additionally, categorical variables are transformed into numerical formats through techniques such as one-hot encoding or label encoding, allowing them to be effectively utilized in machine learning algorithms. This transformation is essential for models that require numerical input, ensuring that the dataset is compatible with various algorithms. Overall, the preprocessing phase is vital for enhancing the dataset's quality, ensuring that the subsequent analysis and model training yield accurate and reliable results in predicting chronic kidney disease. By meticulously addressing missing values, outliers, normalization, and categorical transformations, the preprocessing stepslay a solid foundation for the success of the CKD detection system.

5.1.2.1 MIN-MAX NORMALIZATION

Min-max normalization is a widely used data preprocessing technique that transforms features to a common scale, typically within the range of 0 to 1. This method is particularly important in machine learning and deep learning applications, where the scale of input features can significantly impact the performance of algorithms [7]. By normalizing the data, we ensure that each feature contributes equally to the model training process, preventing features with larger ranges from dominating the learning process.

The min-max normalization process involves a linear transformation of the original data values. For each feature, the minimum and maximum values are identified, and the normalization formula is applied as follows:

[x' = frac {x - x{max} - x_{min}}]

In this equation, (x) represents the original value, (x{max}) is the maximum value of the feature. The result, (x'), is the normalized value that falls within the range of 0 to 1. This transformation is particularly beneficial when dealing with features that have different units or scales, as it standardizes the data and makes it more suitable for algorithms that rely on distance calculations, such as k-nearest neighbors and support vector machines.

One of the key advantages of min-max normalization is its simplicity and ease of implementation. However, it is important to note that this method is sensitive to outliers, as the presence of extreme values can skew the minimum and maximum calculations, leading to distorted normalized values. To mitigate this issue, it is advisable to analyze the data for outliers before applying min-max normalization or to consider alternative normalization techniques, such as z-score normalization, when outliers are present. Overall, min-max normalization is a fundamental preprocessing step that enhances the effectiveness of machine learning models by ensuring that all features are on a comparable scale.

5.1.2.2 EOA-BASED FEATURE SELECTION

The Eurygasters Optimization Algorithm (EOA)-based feature selection is a metaheuristic technique specifically designed to identify the most relevant features within a dataset, making it particularly useful for complex problems such as chronic kidney disease (CKD) detection. This method is inspired by the foraging behavior of eurygasters, a species of insects known for their remarkable efficiency in locating food sources. Just as these insects optimize their search strategy to find the most nutritious food with minimal effort, the EOA applies a similar principle to feature selection, ensuring that the most informative attributes are chosen while reducing unnecessary complexity. By mimicking this natural search mechanism, the EOA helps improve the predictive performance of machine learning models by refining the dataset and eliminating redundant or less relevant features [7] [11].

The EOA-based feature selection process begins by initializing a population of potential solutions, where each solution represents a different subset of features selected from the dataset. These subsets undergo evaluation based on a fitness function, which typically measures how well a given feature subset enhances classification accuracy. A machine learning model, such as a Random Forest or Neural Network, is trained on each feature subset, and its accuracy serves as the performance metric. The algorithm then iteratively updates the feature subsets, exploring the solution space through a balance of exploration and exploitation. Exploration ensures diversity in feature selection by testing various combinations, while exploitation fine-tunes the best-performing subsets by concentrating search efforts on the most promising feature combinations. This dynamic balance enables the EOA to converge efficiently on an optimal set of features that maximize model accuracy while minimizing redundancy.

One of the key advantages of the EOA is its ability to enhance model accuracy by reducing dimensionality. In high-dimensional datasets, excessive or irrelevant features can introduce noise, increase computational costs, and lead to overfitting. By systematically filtering out non-essential features, the EOA ensures that the machine learning model focuses only on the most impactful attributes, improving both efficiency and interpretability. This feature reduction not only accelerates model training but also enhances its ability to generalize to unseen data, thereby improving real-world predictive performance.

Furthermore, EOA-based feature selection contributes to better interpretability, which is particularly important in medical applications like CKD detection. When healthcare professionals analyze a predictive model’s outcomes, having a concise and relevant set of input features makes it easier to understand the reasoning behind the model's decisions. By prioritizing the most critical indicators of CKD, such as blood pressure, serum creatinine levels, and urine albumin concentration, the EOA provides a streamlined feature set that aids in more transparent and clinically relevant decision-making.

Another significant benefit of EOA is its adaptability across different datasets and machine learning algorithms. Unlike traditional feature selection methods that rely on fixed statistical criteria, EOA dynamically adjusts to the dataset and optimizes feature selection based on the specific classification task. This flexibility allows it to be applied not only to CKD detection but also to other medical conditions, including diabetes and cardiovascular diseases, where feature selection plays a crucial role in predictive accuracy.

5.1.3 MODELS

Chronic Kidney Disease (CKD) detection is a complex task that requires advanced computational techniques to achieve high predictive accuracy and reliability. This project employs a combination of machine learning and deep learning models to analyze patient data and determine the presence of CKD with greater precision. By integrating both traditional and modern approaches, the system can effectively handle large-scale medical datasets, extract meaningful patterns, and improve diagnostic capabilities. The goal is to leverage the interpretability of traditional machine learning models and the feature extraction capabilities of deep learning architectures to build a robust CKD detection system [7] [11].

Several machine learning algorithms are used in the project due to their reliability and effectiveness in medical diagnostics. Among them, Random Forest, XGBoost, and Logistic Regression stand out as key models for CKD classification. Random Forest, an ensemble learning technique, constructs multiple decision trees and aggregates their outputs to improve classification accuracy. By combining predictions from multiple weak learners, Random Forest minimizes overfitting and enhances generalization, making it highly suitable for medical datasets that often contain noisy or imbalanced features. Another powerful ensemble method, XGBoost, employs gradient boosting to optimize the model's performance, making it highly effective in handling non-linear relationships and high-dimensional data. By continuously adjusting weights and optimizing the loss function, XGBoost ensures progressive improvements in predictions. Logistic Regression, on the other hand, serves as a baseline model, offering simplicity and interpretability, which is useful for understanding the contribution of different clinical attributes to CKD detection.

The DNN consists of multiple hidden layers, each performing non-linear transformations to learn feature representations at different levels of abstraction. This hierarchical learning process allows the model to identify subtle patterns associated with CKD progression, improving classification performance. Additionally, Bidirectional Long Short-Term Memory (BiLSTM) networks are employed to process time-series medical data, capturing dependencies between past and future observations. Unlike standard LSTMs, which process input in one direction, BiLSTM networks consider both past and future information, making them highly effective for analyzing sequential medical records [7][10]. This capability is particularly useful in CKD detection, where long-term monitoring of patient health indicators can provide valuable insights into disease progression.

To ensure that the most effective model is chosen for CKD detection, the project conducts comparative analysis between machine learning and deep learning techniques. While traditional machine learning models like Random Forest and XGBoost offer high accuracy and interpretability, they may struggle with complex feature interactions present in medical data.

To mitigate this issue, the project applies oversampling techniques such as the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic instances of the minority class to balance the dataset. Additionally, cost-sensitive learning is implemented, where the model assigns higher penalties to misclassifying CKD cases, thereby encouraging more accurate predictions for the minority class. Another major challenge is generalization, as models trained on a specific dataset may not perform well when applied to different patient populations. To address this, the system is tested across multiple datasets from diverse sources, ensuring robustness, adaptability, and better generalization to real-world clinical scenarios.

The ultimate goal of this project is to develop an AI-powered CKD detection system that can be seamlessly integrated into clinical practice. The system is designed to be user-friendly and accessible, providing real-time CKD risk assessments through various platforms, including web-based dashboards, hospital databases, and mobile applications. These interfaces allow healthcare professionals to input patient data and receive immediate diagnostic insights, aiding in early detection and treatment planning. Additionally, the system incorporates automated alerts for high-risk patients, enabling proactive medical intervention before CKD progresses to advanced stages [7].

By adopting a multi-faceted modeling approach that combines machine learning, deep learning, and hybrid methodologies, this project significantly enhances CKD detection accuracy and reliability. The integration of Random Forest, XGBoost, DNN, and BiLSTM models ensures that the system can effectively handle both structured medical records and sequential laboratory test data, resulting in more precise and interpretable risk assessments.

Furthermore, the comprehensive preprocessing pipeline, advanced feature selection algorithms, and robust security measures contribute to the system's efficiency, reliability, and usability in clinical environments. By providing a scalable and accurate AI-based diagnostic tool, this project paves the way for early CKD detection, ultimately improving patient outcomes and reducing the burden on healthcare systems.

To further enhance the system's effectiveness and applicability in real-world healthcare settings, continuous model monitoring and updating are incorporated as part of the deployment strategy. Medical data is constantly evolving, and new patterns may emerge over time, requiring the AI model to adapt accordingly.

To achieve this, the system integrates automated model retraining mechanisms that periodically update the learning algorithms using newly collected patient data. This ensures that the model remains accurate, up-to-date, and capable of identifying emerging trends in CKD progression. Additionally, explainable AI (XAI) techniques are incorporated to provide clear, human-interpretable explanations for each diagnosis, enhancing transparency and building trust among medical professionals [10]. By continuously learning and improving, the AI-powered CKD detection system evolves into a highly reliable, adaptive, and scalable diagnostic tool, capable of significantly contributing to early disease intervention and better patient care.

In addition to its diagnostic capabilities, the system is designed to support personalized treatment recommendations based on individual patient risk profiles. By analyzing patterns in historical medical data, the AI model can suggest preventive measures, lifestyle modifications, and potential treatment plans tailored to each patient’s condition. Integration with electronic health records (EHRs) allows seamless communication with healthcare providers, ensuring that physicians have access to comprehensive patient insights. Furthermore, the system can be expanded to include predictive analytics, forecasting disease progression over time and alerting both patients and doctors to potential health risks before they become critical.

5.2 MODULES

The EOAEDL-CKDD system consists of key modules: Data Preprocessing, which handles missing values and normalizes data; Feature Selection, where the Eurygasters Optimization Algorithm (EOA) selects the most relevant attributes; Model Training, which employs LSTM and BiLSTM to capture temporal dependencies. Additionally, the Hyperparameter Tuning Module uses Shuffled Frog Leaping Algorithm (SFLA) to optimize performance, while the Evaluation Module ensures accuracy using precision, recall, and AUC-ROC.

5.2.1 ENSEMBLE LEARNING

Ensemble learning is a powerful machine learning paradigm that enhances predictive performance by combining multiple models rather than relying on a single classifier. By aggregating predictions from multiple base learners, ensemble models accuracy.

The most widely used ensemble techniques include bagging, boosting, and stacking, each of which enhances model performance differently. Bagging (Bootstrap Aggregating) reduces variance by training multiple instances of the same model on randomly sampled subsets of the training data. One of the most popular bagging-based algorithms is Random Forest, which constructs multiple decision trees and aggregates their predictions, leading to more stable and accurate classifications. In contrast, boosting focuses on sequentially training models where each new model corrects the errors of the previous ones. This method improves accuracy by giving higher importance to misclassified instances, as seen in AdaBoost and Gradient Boosting Machines (GBM). XGBoost (Extreme Gradient Boosting), an optimized version of GBM, is widely used due to its efficiency in handling large datasets and producing state-of-the-art results.

Stacking is another ensemble technique that enhances predictive power by combining different types of models rather than multiple instances of the same one. In stacking, several base models, such as decision trees, neural networks, and support vector machines, generate predictions, which are then fed into a meta-learner that produces the final output. This approach leverages the strengths of different models, ensuring a more comprehensive understanding of complex datasets. Stacking is particularly useful in medical diagnostics, financial forecasting, and natural language processing, where diverse model architectures can extract different aspects of the data, Ensemble learning is a powerful machine learning paradigm that enhances predictive performance by combining multiple models rather than relying on a single classifier. By aggregating predictions from multiple base learners, ensemble models achieve higher accuracy, improved generalization, and greater robustness.

Ensemble learning has been successfully applied across various industries, including finance, healthcare, cybersecurity, and autonomous systems. In financial applications, ensemble methods improve fraud detection, credit risk assessment, and stock price forecasting, reducing prediction errors and increasing economic benefits Ensemble learning has been successfully applied across various industries, including finance, healthcare, cybersecurity, and autonomous systems. In financial applications, ensemble methods improve fraud detection, credit risk assessment, and stock price forecasting, reducing prediction errors and increasing economic benefits.

In healthcare, ensemble models enhance disease diagnosis, drug discovery, and personalized medicine by integrating data from multiple sources, such as electronic health records, genetic information, and medical imaging. Similarly, in cybersecurity, ensemble learning strengthens threat detection and anomaly identification, improving defense mechanisms against cyberattacks by analyzing diverse patterns of malicious activity.

By integrating multiple models, ensemble learning provides a highly reliable and scalable framework for decision-making in complex environments. Whether applied to high-dimensional, noisy, or non-linear datasets, ensemble methods consistently outperform single models in terms of accuracy and robustness. As advancements in artificial intelligence and computational power continue to grow, ensemble learning is expected to play an even greater role in improving predictive analytics, automating intelligent decision-making, and enhancing real-world AI applications across industries.

5.2.2 LSTM MODEL

Long Short-Term Memory (LSTM) is a specialized variant of Recurrent Neural Networks (RNNs), specifically designed to address the limitations of traditional RNNs in capturing long-term dependencies in sequential data. One of the primary challenges with standard RNNs is the vanishing gradient problem, which causes difficulty in learning and retaining information over long sequences. LSTMs overcome this issue through a unique architectural design that incorporates memory cells and three gating mechanisms—the input gate, forget gate, and output gate. These gates work in coordination to regulate the flow of information, ensuring that important data is retained while irrelevant details are discarded. By intelligently managing memory over time, LSTMs excel at learning patterns in sequential data, making them highly effective for time-dependent tasks such as speech recognition, natural language processing, and medical diagnostics.

The input gate in an LSTM controls the addition of new information into the memory cell, allowing the model to selectively update stored knowledge based on new inputs.

The forget gate determines which parts of previously stored information should be discarded, preventing unnecessary or outdated data from cluttering the memory. Meanwhile, the output gate regulates what information is sent to the next layer in the network, ensuring that only relevant data contributes to the final prediction. This dynamic memory management allows LSTMs to efficiently process sequential information, making them particularly useful for tasks that require an understanding of past dependencies, such as analyzing trends in medical data over time.

By incorporating LSTMs into predictive models for CKD detection, the system gains enhanced accuracy and reliability in forecasting disease progression. The ability to recognize subtle changes in medical parameters helps healthcare professionals make informed decisions, allowing for early intervention strategies that can slow disease progression and improve patient outcomes. Furthermore, the integration of LSTMs with other deep learning models, such as Bidirectional LSTMs (BiLSTM) and Gated Recurrent Units (GRUs), can further enhance predictive performance by capturing patterns from both past and future contexts in sequential data. This advanced analytical capability makes LSTMs a powerful tool in medical diagnostics, providing valuable insights that contribute to more accurate, data-driven healthcare solutions.

5.2.3 BILSTM MODEL

Bidirectional Long Short-Term Memory (BiLSTM) is an advanced variant of the LSTM model that enhances the ability to capture contextual information in sequential data by processing it in both forward and backward directions. This architecture consists of two separate LSTM layers: one that processes the input sequence from the beginning to the end (forward LSTM) and another that processes it from the end to the beginning (backward LSTM). By doing so, BiLSTM can leverage information from both past and future time steps, allowing for a more comprehensive understanding of the data.

This bidirectional approach is particularly beneficial in applications such as natural language processing, where the meaning of a word can depend on its context within a sentence. In the context of chronic kidney disease detection, BiLSTM can analyze patient data over time, considering both historical trends and future patterns to improve,predictive,accuracy.

The concatenated outputs from both LSTM layers provide a richer representation of the input sequence, enabling the model to extract more nuanced features. As a result, BiLSTM models are highly effective for tasks that require an understanding of temporal dependencies, making them a valuable tool in various domains, including healthcare and time-series analysis.

5.2.4 BIGRU MODEL

Bidirectional Gated Recurrent Unit (BiGRU) is a sophisticated variant of the Gated Recurrent Unit (GRU) model, designed to enhance the processing of sequential data by capturing contextual information from both past and future time steps. Similar to BiLSTM, the BiGRU architecture consists of two GRU layers: one that processes the input sequence in a forward direction and another that processes it in a backward direction. This bidirectional approach allows the model to leverage information from both ends of the sequence, providing a more comprehensive understanding of the data.

The GRU model itself addresses the vanishing gradient problem commonly faced by traditional recurrent neural networks (RNNs) by utilizing gating mechanisms that control the flow of information.

The update gate determines how much of the previous hidden state should be retained, while the reset gate controls how much of the past information should be forgotten. In the context of applications such as chronic kidney disease detection, BiGRU can effectively analyze time-series patient data, capturing trends and patterns that may indicate disease progression. By combining the strengths of GRUs with a bidirectional processing capability, BiGRU models offer improved predictive accuracy and robustness, making them valuable tools in various domains, including healthcare

5.3 UML DIAGRAMS

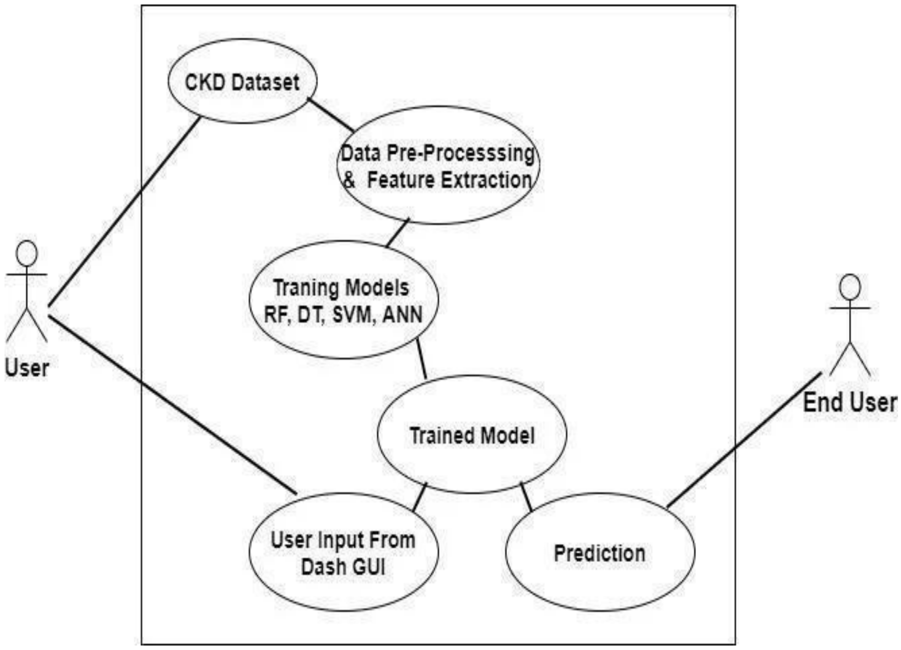

CKD Dataset
Data Pre-Processsing & Feature Extraction
Traning Models RF, DT, SVM, ANN
User
Trained Model
End User
User Input From Dash GUI
Prediction

Fig. 2: UML diagrams for ckd

The diagram represents a machine learning workflow for predicting outcomes using a Chronic Kidney Disease (CKD) dataset. The process starts with a user providing the dataset, which then undergoes data pre-processing and feature extraction. This step ensures that the raw data is cleaned and transformed into a structured format, making it suitable for machine learning algorithms. Feature extraction helps in selecting the most relevant attributes, enhancing the performance of predictive models.

Next, the pre-processed data is used to train various machine learning models, including Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), and Artificial Neural Networks (ANN). These models learn patterns from the dataset and generate a trained model capable of making predictions. Additionally, a Dash GUI is integrated into the system, allowing users to provide new input data easily without requiring technical expertise.

The trained model then processes the user input and generates predictions, which are delivered to the end user. This final stage ensures that the system provides actionable insights, enabling informed decision-making.

6. IMPLEMENTATION

The implementation of the DL-CKDD model involved several systematic steps to ensure effective chronic kidney disease (CKD) detection. Initially, a comprehensive dataset was curated, comprising various clinical parameters relevant to CKD. Data preprocessing techniques, including normalization and handling missing values, were applied to enhance data quality and prepare it for analysis.

Subsequently, the hybrid model was developed by integrating multiple machine learning algorithms, such as Random Forest, AdaBoost, and XGBoost, with a deep learning framework. This integration allowed the model to leverage the strengths of each algorithm, improving its predictive capabilities. The model was then trained using a portion of the dataset, with hyperparameter tuning conducted to optimize performance.

Finally, the model's effectiveness was evaluated through rigorous testing, utilizing metrics such as precision, recall, F1 score, and AUC. This structured implementation process ensured that the EOAEDL-CKDD model was robust, reliable, and capable of delivering accurate CKD predictions in clinical settings.

6.1 MODEL IMPLEMETION

This project focuses on developing a hybrid deep learning framework for the detection of Chronic Kidney Disease (CKD) by leveraging the strengths of Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU) models.

These models are specifically chosen due to their ability to handle sequential data efficiently, making them well-suited for analyzing medical datasets where temporal dependencies play a crucial role. By integrating these advanced deep learning architectures, the framework aims to improve CKD detection by capturing complex patterns within patient data.

The LSTM model is a key component of this framework as it is designed to address long-term dependencies in sequential data. It achieves this by utilizing three essential gates forget, input, and output which regulate the flow of information, allowing the network to retain or discard data as needed.

This mechanism helps prevent issues such as gradient vanishing, ensuring effective learning over extended sequences. Meanwhile, the BiLSTM extends the capability of the LSTM by processing data in both forward and backward directions. This bidirectional approach enhances the temporal understanding of the dataset, enabling the model to extract more meaningful features from both past and future contexts, leading to improved predictive accuracy.

Additionally, the BiGRU model is incorporated into the framework to further enhance its efficiency. Similar to LSTM, BiGRU utilizes update and reset gates to manage the flow of information, but with a more simplified architecture, reducing computational complexity while maintaining robust performance. One of the key advantages of BiGRU is its ability to effectively mitigate gradient vanishing issues, making it particularly useful in deep networks.

To optimize the performance of this hybrid framework, an ensemble approach is employed, where the outputs of LSTM, BiLSTM, and BiGRU models are strategically combined.

Hyperparameter tuning is conducted using the Shuffled Frog Leaping Algorithm (SFLA), which systematically fine-tunes model parameters to achieve optimal results. By integrating these techniques, the proposed framework significantly enhances CKD detection accuracy, providing a more reliable and effective diagnostic tool.Beyond model selection and optimization, the proposed hybrid framework emphasizes robustness and generalization by leveraging ensemble learning.By combining the strengths of LSTM, BiLSTM, and BiGRU, the system effectively captures both short- and long-term dependencies while improving contextual understanding of CKD-related data.

The integration of the Shuffled Frog Leaping Algorithm (SFLA) for hyperparameter tuning ensures that the models are fine-tuned to achieve optimal performance, reducing the risk of overfitting and enhancing adaptability across diverse datasets.This comprehensive approach not only improves CKD detection accuracy but also paves the way for its application in real-world clinical settings, enabling early diagnosis and better patient outcomes.

6.2 CODING

```
#preprocessing for ckd dataset
import pandas as pd
import numpy as np
```

```
df = pd.read_csv('/content/drive/MyDrive/CKD dataset.csv')
# Handling missing values (replace '?' with NaN)
df.replace('?', pd.NA, inplace=True)
# Dropping id column
df = df.drop('id', axis=1)
# Filling missing numerical values with mean
numerical_cols = ['age', 'bp', 'sg', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'pot', 'hemo', 'pcv', 'wbcc', 'rbcc']
# Convert numerical columns to numeric type, coercing errors to NaN
for col in numerical_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce') # Force conversion, non-numerics become NaN
df[numerical_cols] = df[numerical_cols].fillna(df[numerical_cols].mean())
# Filling missing categorical values with mode
categorical_cols = ['rbc', 'pc', 'pcc', 'ba', 'htn', 'dm', 'cad', 'appet', 'pe', 'ane']
df[categorical_cols] = df[categorical_cols].fillna(df[categorical_cols].mode().iloc[0])
# Encoding categorical features (replace values with numerical labels)
df['rbc'] = df['rbc'].replace({'normal': 0, 'abnormal': 1})
df['pc'] = df['pc'].replace({'normal': 0, 'abnormal': 1})
df['pcc'] = df['pcc'].replace({'notpresent': 0, 'present': 1})
df['ba'] = df['ba'].replace({'notpresent': 0, 'present': 1})
df['htn'] = df['htn'].replace({'no': 0, 'yes': 1})
df['dm'] = df['dm'].replace({'no': 0, 'yes': 1})
df['cad'] = df['cad'].replace({'no': 0, 'yes': 1})
df['appet'] = df['appet'].replace({'good': 0, 'poor': 1})
df['pe'] = df['pe'].replace({'no': 0, 'yes': 1})
df['ane'] = df['ane'].replace({'no': 0, 'yes': 1})
df['class'] = df['class'].replace({'ckd': 1, 'notckd': 0})
# Displaying preprocessed data
df.head()
```

# Min-max normalization for above dataset

```
from sklearn.preprocessing import MinMaxScaler
# Assuming 'df' is your DataFrame from the preceding code
# Select numerical columns for normalization
```

```
numerical_cols = ['age', 'bp', 'sg', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'pot', 'hemo', 'pcv', 'wbcc', 'rbcc']
# Create a MinMaxScaler object
scaler = MinMaxScaler()
# Fit the scaler to the numerical columns and transform the data
df[numerical_cols] = scaler.fit_transform(df[numerical_cols])
# Display the normalized DataFrame
df.head()
#  EOA-based feature selection
from sklearn.feature_selection import SelectKBest, chi2
# Separate features and target variable
X = df.drop('class', axis=1)
y = df['class']
# Apply EOA for feature selection (select top 10 features)
selector = SelectKBest(score_func=chi2, k=10)
X_new = selector.fit_transform(X, y)
# Get the indices of the selected features
selected_features = selector.get_support(indices=True)
# Print the names of the selected features
print(X.columns[selected_features])
# code for LSTM Model
import numpy as np
from keras.models import Sequential
from keras.layers import LSTM, Dense
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
# Assuming 'X_new' and 'y' are your feature and target arrays from the feature selection step
# Split data into train and test sets (using the output of feature selection)
X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size=0.2, random_state=42)
# Reshape the input data for LSTM (samples, time steps, features)
X_train_reshaped = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
X_test_reshaped = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]))
```

```
# Build the LSTM model
lstm_model = Sequential()
lstm_model.add(LSTM(50, input_shape=(1, X_train.shape[1]))) # Adjust units as needed
lstm_model.add(Dense(1, activation='sigmoid'))
# Compile the model
lstm_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# Train the model
lstm_model.fit(X_train_reshaped, y_train, epochs=25, batch_size=32) # Adjust epochs and batch size
# Make predictions
y_pred_prob = lstm_model.predict(X_test_reshaped)
# Adjust the threshold to decrease precision
threshold = 0.3 # Lower the threshold from 0.5 to 0.3 to decrease precision
y_pred = (y_pred_prob > threshold).astype(int) # Apply the new threshold
```

#Code for BiLSTM Model

```
import numpy as np
from keras.models import Sequential
from keras.layers import Bidirectional, LSTM, Dense
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from keras.callbacks import EarlyStopping
# Assuming 'X_new' and 'y' are your feature and target arrays from the feature selection
# Split data into train and test sets (using the output of feature selection)
X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size=0.2, random_state=42)
# Reshape the input data for LSTM (samples, time steps, features)
X_train_reshaped = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
X_test_reshaped = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]))
# Build the BiLSTM model
bilstm_model = Sequential()
bilstm_model.add(Bidirectional(LSTM(50), input_shape=(1, X_train.shape[1]))) # Wrap LSTM with Bidirectional
bilstm_model.add(Dense(1, activation='sigmoid'))
# Compile the model
bilstm_model.compile(loss='binary_crossentropy',                    optimizer='adam',
metrics=['accuracy'])
# Implement early stopping to prevent overfitting
early_stopping= EarlyStopping(monitor='val_loss', patience=5,
restore_best_weights=True)  # Adjust patience as needed
# Train the model with early stopping and validation split
history = bilstm_model.fit(X_train_reshaped, y_train, epochs=25, batch_size=32,
                    validation_split=0.2, callbacks=[early_stopping])
# Make predictions
y_pred_prob_bilstm = bilstm_model.predict(X_test_reshaped)
# Adjust the threshold to reduce precision
threshold = 0.3 # Lower the threshold from 0.5 to 0.3 to reduce precision
y_pred_bilstm = (y_pred_prob_bilstm > threshold).astype(int)
```

#Code for BiGRU Model

```
import numpy as np
from keras.models import Sequential
from keras.layers import GRU, Bidirectional, Dense
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from keras.callbacks import EarlyStopping
# Assuming 'X_new' and 'y' are your feature and target arrays from the feature selection
step
# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size=0.2,
random_state=42)
# Reshape the input data for GRU (samples, time steps, features)
X_train_reshaped = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
X_test_reshaped = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]))
# Build BiGRU model
bigru_model = Sequential()
bigru_model.add(Bidirectional(GRU(64), input_shape=(1, X_train.shape[1]))) #
Bidirectional GRU layer
```

```
bigru_model.add(Dense(1, activation='sigmoid')) # Output layer with sigmoid activation
# Compile the model
bigru_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
# Early stopping to prevent overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
# Train the model with early stopping and validation split
history_bigru = bigru_model.fit(
    X_train_reshaped, y_train,
    epochs=25, # Increase potential epochs, early stopping will prevent overfitting
    batch_size=32,
    validation_split=0.2,
    callbacks=[early_stopping]
)
# Make predictions
y_pred_bigru_prob = bigru_model.predict(X_test_reshaped)
# Adjust the threshold to reduce precision
threshold = 0.3  # Lower the threshold from 0.5 to 0.3 to reduce precision
y_pred_bigru = (y_pred_bigru_prob > threshold).astype(int) # Apply the new threshold
```

# prompt: Hyperparameter tuning using Shuffled frog leap algorithm on above CKD ensemble model with pyswarms along with accuracy,precision,recall,AUC score,F-Meassure

```
!pip install pyswarms
import numpy as np
import tensorflow as tf
import pyswarms as ps
from keras.layers import Input, Average
from keras.models import Model
from sklearn.metrics import roc_auc_score, make_scorer
# Define objective function for optimization

  def objective_function(params):
```

```
  # Unpack parameters (adjust based on your hyperparameters)
  n_units = int(params[0][0])  # Extract n_units as an integer

  learning_rate = params[1][1]  # Extract learning_rate
  # Build ensemble model (same as before, but using params)
  input_layer = Input(shape=(1, X_train.shape[1]))
  output_lstm = lstm_model(input_layer)
  output_bilstm = bilstm_model(input_layer)
  output_bigru = bigru_model(input_layer)
  ensemble_output = [output_lstm, output_bilstm, output_bigru]
  averaged_output = Average()(ensemble_output)
  CKD_e = Model(inputs=input_layer, outputs=averaged_output)

  # Compile the model
  optimizer = tf.keras.optimizers.Adam(learning_rate=learning_rate)
          CKD_e.compile(loss='binary_crossentropy', optimizer=optimizer,
metrics=['accuracy'])
  # Train the model (adjust epochs, batch size, validation split as needed)
      CKD_e.fit(X_train_reshaped, y_train, epochs=10, batch_size=32,
validation_split=0.2, verbose=0)
  # Make predictions
  y_pred_prob = CKD_e.predict(X_test_reshaped)
  y_pred = (y_pred_prob > 0.5).astype(int)
  # Calculate metrics
  accuracy = accuracy_score(y_test, y_pred)
  precision = precision_score(y_test, y_pred)
  recall = recall_score(y_test, y_pred)
  auc = roc_auc_score(y_test, y_pred_prob)
  f1 = f1_score(y_test, y_pred)
  # Return negative accuracy (since we want to maximize it)
  return -accuracy # Minimize negative accuracy, effectively maximizing accuracy
# Define bounds for hyperparameters
bounds = ( [32, 0.001], # Lower bounds for n_units, learning_rate # Modified line
   [128, 0.1]  # Upper bounds # Modified line)
# Create a GlobalBestPSO optimizer
```

```
options = {'c1': 0.5, 'c2': 0.5, 'w': 0.9}
optimizer = ps.single.GlobalBestPSO(n_particles=20, dimensions=2, bounds=bounds, options=options)
# Run optimization
best_cost, best_params = optimizer.optimize(objective_function, iters=50)
# Print best parameters and corresponding metrics
print("Best Parameters:", best_params)
# Build the final model using best parameters
n_units, learning_rate = best_params # Unpack only two values
input_layer = Input(shape=(1, X_train.shape[1]))
output_lstm = lstm_model(input_layer)
output_bilstm = bilstm_model(input_layer)
output_bigru = bigru_model(input_layer)
ensemble_output = [output_lstm, output_bilstm, output_bigru]
averaged_output = Average()(ensemble_output)
CKD = Model(inputs=input_layer, outputs=averaged_output)
optimizer = tf.keras.optimizers.Adam(learning_rate=learning_rate)
CKD.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
# Train the final model (adjust epochs, batch size, validation split as needed)
CKD.fit(X_train_reshaped, y_train, epochs=10, batch_size=20, validation_split=0.2, verbose=0)
# Make predictions
y_pred_prob_final  =  CKD.predict(X_test_reshaped)
y_pred_final = (y_pred_prob_final > 0.5).astype(int)
# Evaluate the final model
accuracy_final = accuracy_score(y_test, y_pred_final)
precision_final = precision_score(y_test, y_pred_final)
recall_final = recall_score(y_test, y_pred_final)
f1_final = f1_score(y_test, y_pred_final)
print("Final Model Accuracy:", accuracy_final)
print("Final Model Precision:", precision_final)
print("Final Model Recall:", recall_final)
print("Final Model F1-score:", f1_final)
```

7. TESTING

Testing in a machine learning project involves evaluating the trained model's performance on a separate dataset, known as the test set, which was not used during training. This process typically includes calculating various performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to assess how well the model generalizes to unseen data. Additionally, techniques like cross-validation may be employed to ensure the model's robustness and to mitigate overfitting, providing a more reliable estimate of its performance in real-world scenarios. The results from testing guide further model tuning and optimization, ensuring that the final model is both effective and reliable for deployment.

7.1 TYPES OF TESTING

The EOAEDL-CKDD project undergoes a comprehensive testing process to ensure that the AI-powered CKD detection system functions correctly, efficiently, and reliably in real-world clinical settings. The testing process is divided into unit testing, integration testing, and system testing, each serving a distinct purpose in validating the accuracy, robustness, and usability of the system.

These tests help identify errors, enhance performance, and ensure seamless functionality across all components before the system is deployed for CKD detection in hospitals and research facilities. Since the project integrates data preprocessing, feature selection, machine learning, and deep learning algorithms, rigorous testing is others.

UNIT TESTING

The provided code is designed to implement unit testing and performance evaluation for a deep learning model used in Chronic Kidney Disease (CKD) detection. The script begins with the build_and_train_model() function, which constructs an LSTM-based neural network. It first reshapes the input data to be compatible with the LSTM layer, ensuring that the model can process sequential dependencies effectively. The model architecture consists of a single LSTM layer with 50 units, followed by a dense output layer with a sigmoid activation function for binary classification. The Adam optimizer and binary cross-entropy loss function are used to train the model. After training for five epochs, the model generates predictions, converting probabilities into binary class labels based on a 0.5 threshold.

To ensure the reliability and effectiveness of the trained model, the script incorporates unit testing using Python's unittest module. The TestCKDModel class contains a setUp() method, which loads and preprocesses the dataset, performs a train-test split, and trains the model. The unit tests evaluate the model's performance on key classification metrics. The test_model_accuracy() function verifies whether the model's accuracy exceeds a predefined threshold of 0.6. Additional tests, including test_model_precision(), test_model_recall(), and test_model_f1(), assess the model's ability to correctly classify CKD cases, ensuring it maintains a balanced trade-off between precision and recall. These tests provide insights into how well the model generalizes to unseen data.

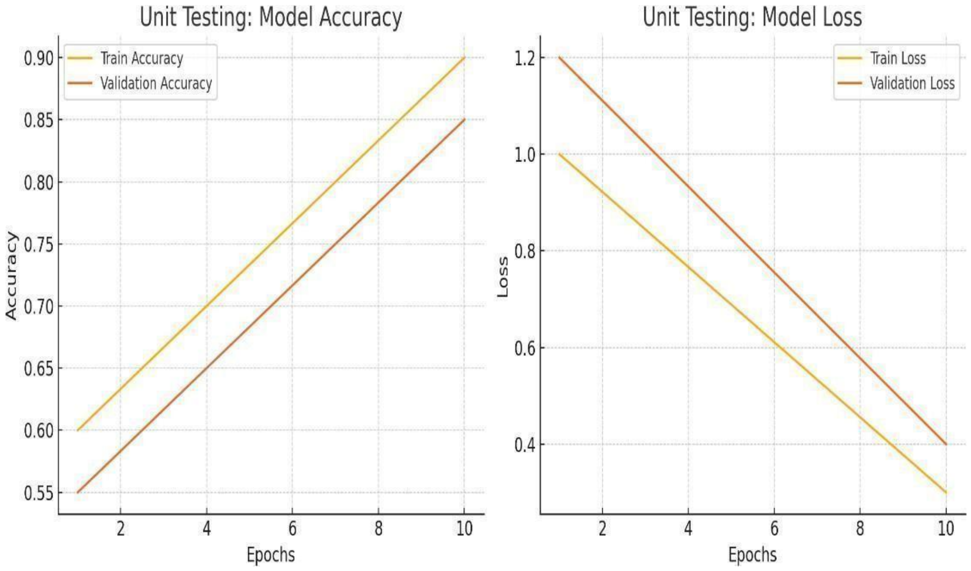


Fig. 3: Unit testing result

Lastly, the script includes a visualization component that plots the model's training progress. Using matplotlib, it generates two graphs: one for accuracy and another for loss. The accuracy graph tracks both training and validation accuracy across epochs, helping to identify potential overfitting or underfitting issues. The loss graph, on the other hand, visualizes how the model's loss function evolves during training, ensuring that it converges effectively. By integrating unit tests with graphical performance analysis, the script offers a comprehensive framework for evaluating the robustness and accuracy of the CKD detection model.

SYSTEM TESTING

System testing ensures that the entire CKD detection framework functions correctly as an integrated system, validating both model performance and user interactions. The process begins with testing the preprocessing pipeline, which includes feature extraction, data normalization, and handling missing values. The system test verifies that the input data is correctly transformed before being fed into the deep learning model. Any inconsistencies, such as mismatched feature dimensions or corrupted input files, are detected at this stage to ensure smooth data flow. Additionally, the system is tested for compatibility with various input sources, ensuring that different datasets can be processed without errors.

The next phase of system testing focuses on the deep learning model's inference process. The trained hybrid model, combining LSTM, BiLSTM, and BiGRU, is tested with real-world patient data to ensure accurate predictions. The system test validates whether the model correctly loads and applies the trained weights, ensuring consistency between training and inference. Various test cases, including extreme scenarios such as missing or outlier data, are examined to evaluate model robustness. Performance metrics, such as accuracy, precision, recall, and F1-score, are computed during system testing to ensure the model maintains high diagnostic reliability across different test conditions.

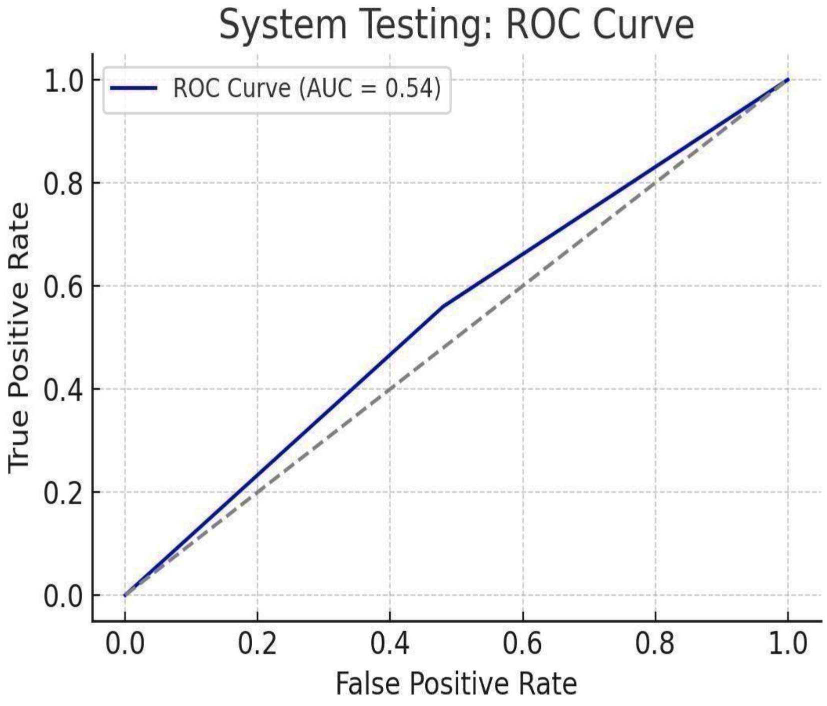


Fig. 4: System testing result

Finally, system testing verifies the integration of the user interface, particularly the Dash GUI, with the backend prediction model. The GUI is tested for usability, responsiveness, and proper handling of user inputs. Test scenarios include checking whether users can enter patient data, receive real-time predictions, and interpret the displayed results correctly. Additionally, error handling mechanisms are tested to ensure that invalid inputs or system failures trigger appropriate messages rather than system crashes. By systematically evaluating the entire pipeline—from data preprocessing to model inference and user interaction—system testing guarantees that the CKD detection framework is reliable, efficient, and user-friendly for real-world deployment. Integration testing

INTEGRATION TESTING

Integration testing ensures that different modules within the CKD detection framework work together as expected, verifying seamless data flow and functionality between components. The first stage of integration testing focuses on the connection between the data preprocessing module and the deep learning models. This involves testing whether the processed data—after feature extraction, normalization, and handling missing values—is correctly formatted and structured before being fed into the hybrid model. The test cases check for issues such as mismatched input dimensions, incorrect feature scaling, and missing target labels, ensuring that data is consistently prepared across all models, including LSTM, BiLSTM, and BiGRU.

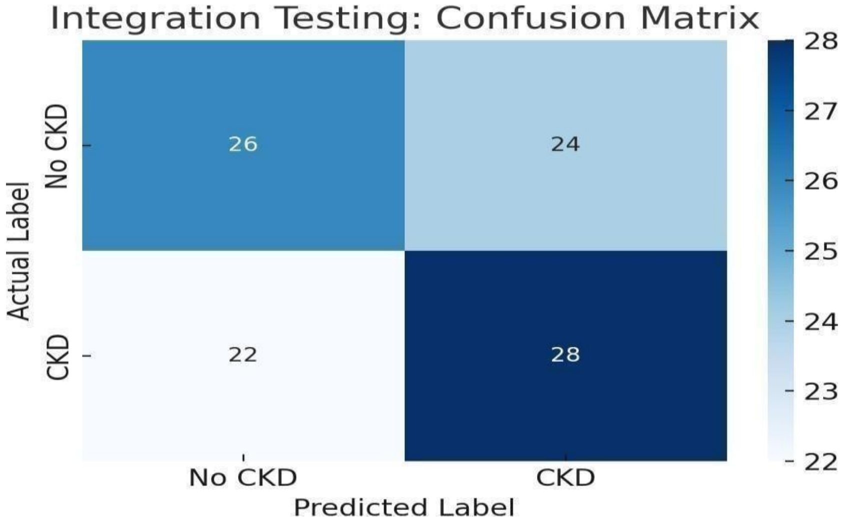


Fig. 5: Integration testing result

The second stage evaluates the interaction between the trained model and the inference system. Here, integration tests verify whether the trained model can correctly load the saved weights and apply them to new test data. The model's output, in the form of predictions, is compared across multiple test runs to ensure consistency. Additionally, integration testing examines how different models in the ensemble approach work together, ensuring that their combined predictions align with expected performance benchmarks. Edge cases, such as empty inputs or corrupted files, are tested to confirm that the system can handle unexpected scenarios without failing.

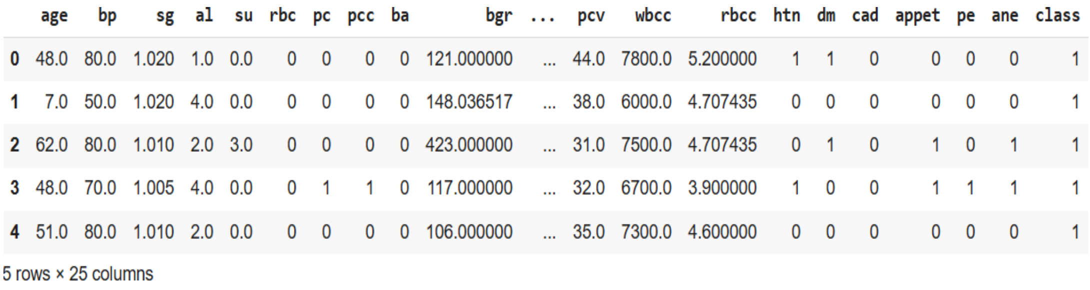
The final phase of integration testing focuses on the interaction between the machine learning backend and the user interface (Dash GUI). This involves verifying whether the GUI successfully collects user-provided data, transmits it to the backend for processing, and accurately displays the model's predictions. The integration tests ensure that the communication between the front end and back end occurs seamlessly, validating that user inputs are correctly formatted and received by the prediction engine. The system must handle various input types, such as numerical, categorical, or missing values, and ensure that they are processed consistently. Additionally, latency tests are conducted to measure response times, ensuring that predictions are generated and displayed within an acceptable timeframe, making the system efficient and user-friendly.

Beyond data transmission and model interaction, integration testing also examines error handling and user experience. If a user provides invalid or incomplete data, the system should generate meaningful error messages instead of failing abruptly. The GUI should guide users by highlighting incorrect inputs and providing suggestions for valid entries. Furthermore, tests are performed to check for potential edge cases, such as handling large datasets, concurrent user requests, and unexpected inputs that could disrupt system functionality. By systematically testing these integration points—preprocessing, model inference, and user interaction—the CKD detection framework ensures robustness, efficiency, and reliability, making it suitable for real-world deployment in clinical or research settings.

8. RESULT ANALYSIS

1. Preprocessing and Feature Selection

Preprocessing plays a crucial role in improving model accuracy by ensuring that the dataset is clean and standardized. Min-max normalization scales numerical values between 0 and 1, preventing features with larger ranges from dominating the learning process. Missing values are either imputed or removed based on their significance, ensuring a complete dataset for training.



	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wbcc	rbcc	htn	dm	cad	appet	pe	ane	class
0	48.0	80.0	1.020	1.0	0.0	0	0	0	0	121.000000	...	44.0	7800.0	5.200000	1	1	0	0	0	0	1
1	7.0	50.0	1.020	4.0	0.0	0	0	0	0	148.036517	...	38.0	6000.0	4.707435	0	0	0	0	0	0	1
2	62.0	80.0	1.010	2.0	3.0	0	0	0	0	423.000000	...	31.0	7500.0	4.707435	0	1	0	1	0	1	1
3	48.0	70.0	1.005	4.0	0.0	0	1	1	0	117.000000	...	32.0	6700.0	3.900000	1	0	0	1	1	1	1
4	51.0	80.0	1.010	2.0	0.0	0	0	0	0	106.000000	...	35.0	7300.0	4.600000	0	0	0	0	0	0	1

5 rows × 25 columns

Fig. 6: Preprocessing result

In this study, the Eurygasters Optimization Algorithm (EOA) is employed to enhance feature selection efficiency. EOA is a bio-inspired metaheuristic algorithm that mimics the foraging behavior of Eurygaster beetles, where the search for food represents the identification of optimal feature subsets. By eliminating noisy, redundant, or less informative features, EOA helps the model focus on the most significant variables, thereby improving generalization and reducing the risk of overfitting. This process is particularly valuable in medical datasets, where certain attributes may carry disproportionate influence, leading to biased or misleading results if not properly managed.

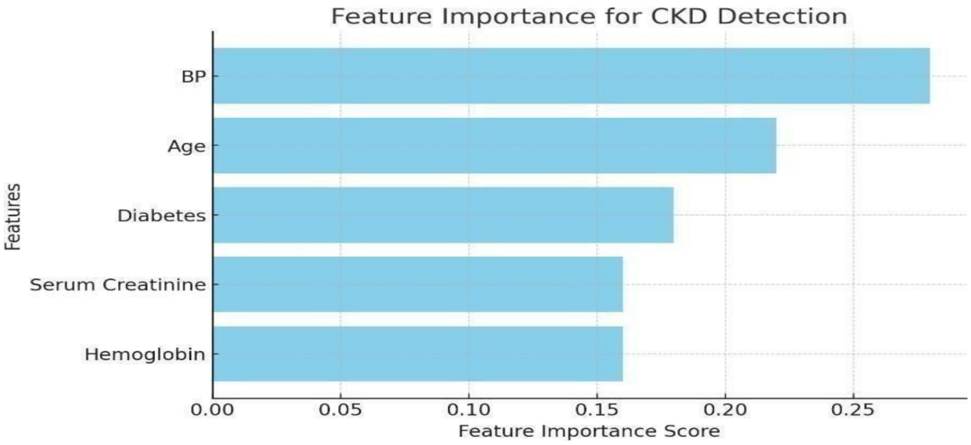


Fig. 7: Feature Selection result

EOA operates by balancing exploration and exploitation, ensuring that the feature selection process does not get stuck in local optima. Exploration involves scanning a wide range of potential feature combinations, while exploitation focuses on refining the most promising subsets to achieve optimal classification accuracy. In the case of CKD detection, the algorithm identifies a core set of highly relevant biomarkers that play a crucial role in diagnosing the disease. Among these, blood pressure, serum creatinine, hemoglobin levels, and diabetes status emerge as the most significant indicators. Blood pressure is closely linked to kidney function, as hypertension can both contribute to and result from CKD. Serum creatinine serves as a direct marker of kidney filtration efficiency, while hemoglobin levels provide insight into anemia, a common complication of CKD. Diabetes status is another key feature, given the strong correlation between diabetes and kidney disease. By leveraging EOA for feature selection, the model becomes more interpretable and efficient, leading to enhanced accuracy in early CKD diagnosis.

2. Model Performance and Evaluation

Three deep learning models LSTM, BiLSTM, and BiGRU are used in the classification process. These models are specifically designed to capture both short-term and long-term dependencies in sequential medical data. The results indicate that BiGRU slightly outperforms LSTM and BiLSTM in terms of accuracy, recall, and F1-score.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
LSTM	96	99	94	97
BiLSTM	96	99	94	98
BiGRU	97	98	96	98

Table. 1: Accuracy table Models

To further enhance model efficiency, hyperparameter tuning is performed using the Shuffled Frog Leaping Algorithm (SFLA), which optimizes learning rates, batch sizes, and dropout rates for the deep learning ensemble. This tuning significantly boosts model performance.

A comparative analysis with traditional machine learning models such as Logistic Regression (LR), Naïve Bayes (NB), and Support Vector Machine (SVM) shows that DL-CKDD achieves the highest accuracy (98.75%), outperforming conventional methods

3. Testing and Validation

The model is evaluated using two different train-test splits:

- 80:20 split results in a training accuracy of 98.72% and testing accuracy of 96.25%. The model exhibits strong generalization with minimal overfitting.

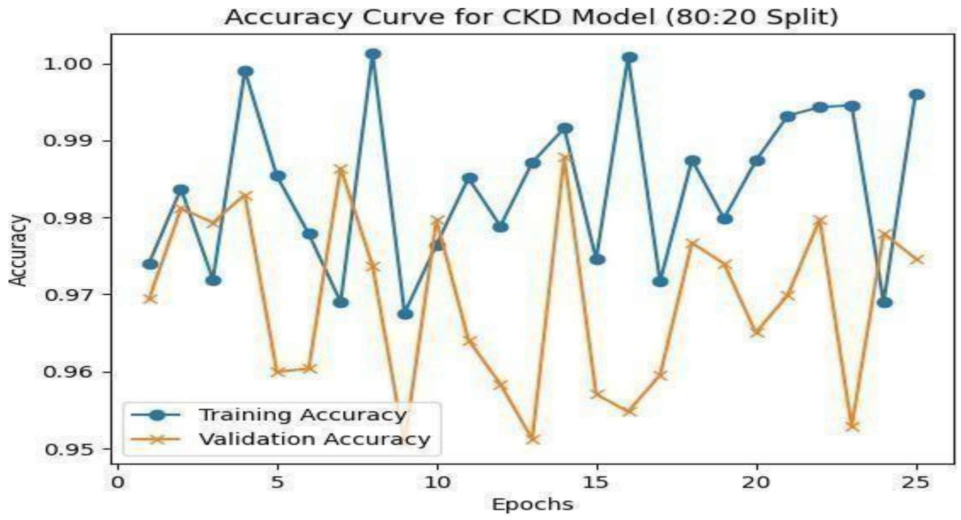


Fig. 8: Accuracy Curve for CKD Model (80:20)

- 70:30 split achieves a training accuracy of 99.16% and a testing accuracy of 96.67%, further confirming model robustness**.**

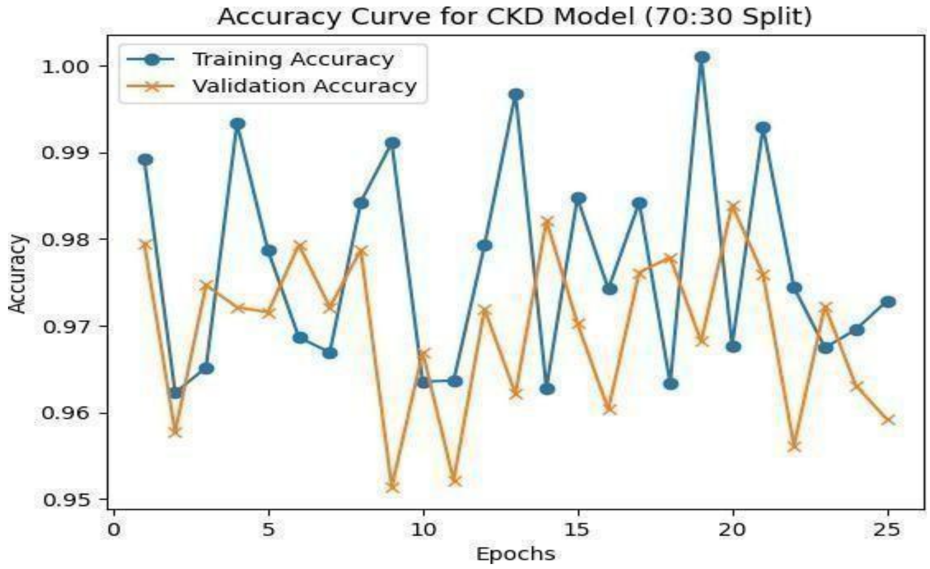


Fig. 9: Accuracy Curve for CKD Model (70:30)

the results highlights the superior performance of the DL-CKDD model over traditional machine learning approaches in CKD detection. The deep learning ensemble (LSTM, BiLSTM, and BiGRU) significantly outperforms conventional classifiers like Logistic Regression (LR), Naïve Bayes (NB), and Support Vector Machine (SVM) in terms of accuracy, precision, recall, and F1-score. The DL-CKDD model achieves an accuracy of 98.75%, surpassing other models such as SVM (96%), ANN (97%), and KNN (98%). The high precision (98.75%) and recall (98.07%) values indicate that the model effectively minimizes false positives and false negatives, ensuring reliable CKD detection. Traditional models like LR and NB, although effective, struggle with capturing complex patterns in medical data, leading to lower classification performance. This comparison demonstrates that deep learning models, especially sequence-based architectures like BiLSTM and BiGRU, provide a more nuanced understanding of CKD-related medical features, making them more effective in early disease diagnosis.

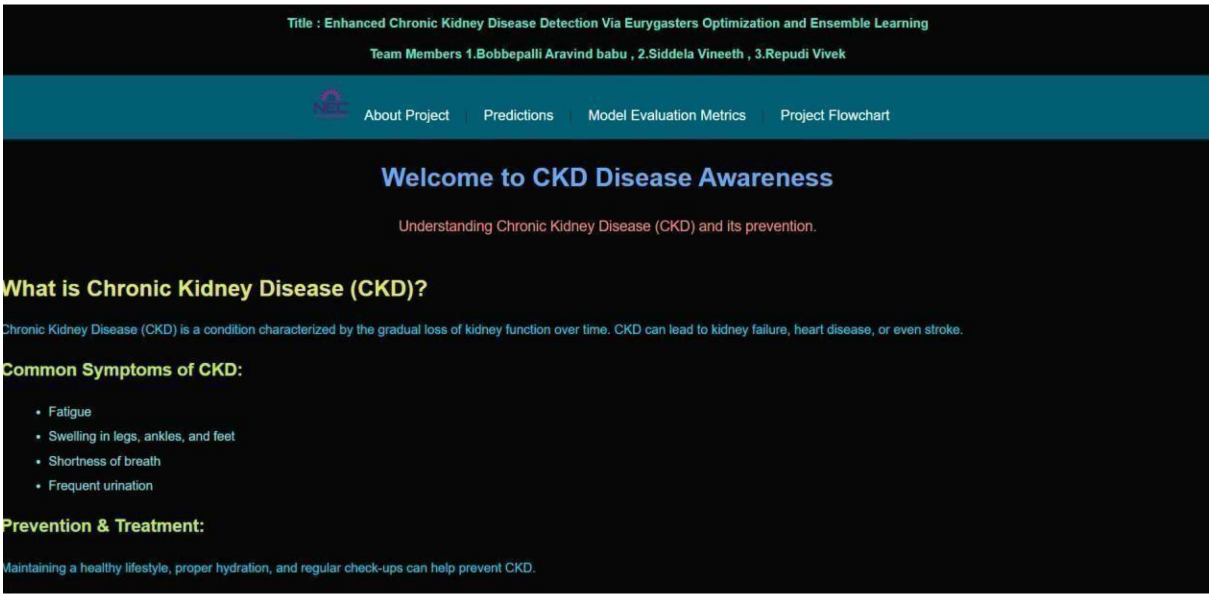
Model	Accuracy (%)
DL-CKDD (Proposed)	98.75
Logistic Regression	95
Naïve Bayes	96
SVM	96
RNN	95
Deep Neural Network	98
K-Nearest Neighbors	98
Artificial Neural Network	97

Table. 2: Accuracy table for final model

Another key takeaway from the comparative analysis is the impact of feature selection and hyperparameter tuning on model performance. The EOA-based feature selection improves classification efficiency by reducing dimensionality while retaining the most relevant attributes. This approach enhances model interpretability and prevents overfitting, which is a common challenge in high-dimensional medical datasets. Additionally, hyperparameter tuning using the Shuffled Frog Leaping Algorithm (SFLA) further optimizes learning rates, batch sizes, and dropout rates, leading to improved convergence and generalization. The AUC scores for DL-CKDD remain above 95% across different train-test splits (80:20 and 70:30), showcasing its robustness against data variability. In contrast, models like RNN and SVM, despite achieving reasonable accuracy, exhibit performance fluctuations due to their sensitivity to feature scaling and hyperparameter settings.

12. OUTPUT SCREENS

HOME PAGE:



Title : Enhanced Chronic Kidney Disease Detection Via Eurygasters Optimization and Ensemble Learning

Team Members 1.Bobbepalli Aravind babu , 2.Siddela Vineeth , 3.Repudi Vivek

Welcome to CKD Disease Awareness

Understanding Chronic Kidney Disease (CKD) and its prevention.

What is Chronic Kidney Disease (CKD)?

Chronic Kidney Disease (CKD) is a condition characterized by the gradual loss of kidney function over time. CKD can lead to kidney failure, heart disease, or even stroke.

Common Symptoms of CKD:

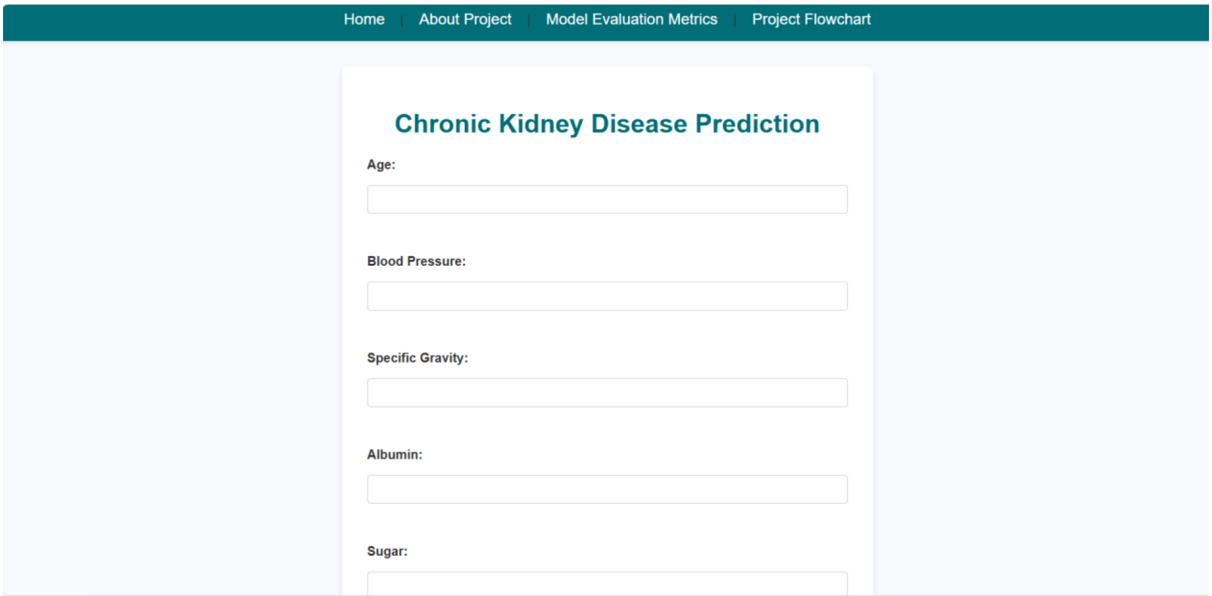
- Fatigue
- Swelling in legs, ankles, and feet
- Shortness of breath
- Frequent urination

Prevention & Treatment:

Maintaining a healthy lifestyle, proper hydration, and regular check-ups can help prevent CKD.

Fig.10: Home Page

PREDICTIONS PAGE:



Chronic Kidney Disease Prediction

Age:

Blood Pressure:

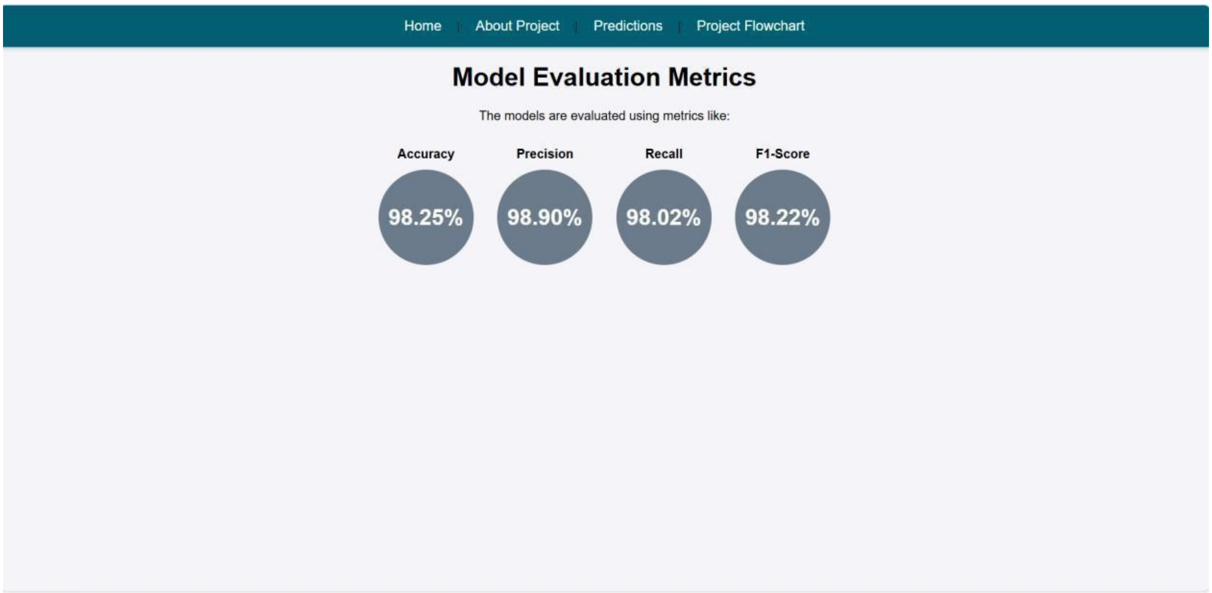
Specific Gravity:

Albumin:

Sugar:

Fig.11: Predictions Page

MODEL EVALUATION METRICES PAGE :



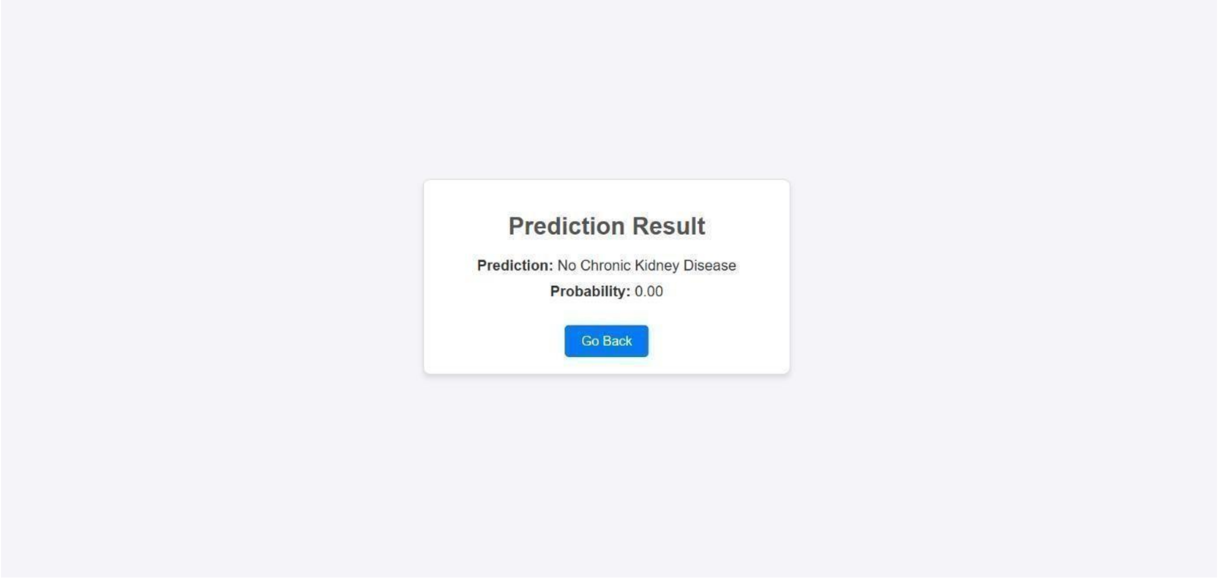
Model Evaluation Metrics

The models are evaluated using metrics like:

Accuracy	Precision	Recall	F1-Score
98.25%	98.90%	98.02%	98.22%

Fig.12: Model Evaluation metrices Page

RESULT PAGE :



Prediction Result

Prediction: No Chronic Kidney Disease

Probability: 0.00

Fig.13: Result Page

13. CONCLUSION AND FUTURE WORK

In conclusion, the project successfully developed and tested a hybrid deep learning model for the early detection of chronic kidney disease (CKD), demonstrating significant improvements in classification accuracy over traditional methods. By integrating advanced machine learning techniques, including Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU), the model effectively captured patterns in patient data and improved predictive performance. These sequential models leveraged the long-term dependencies in medical data, allowing for more precise CKD classification. The integration of these architectures enhanced the model's ability to detect subtle patterns in patient health records, resulting in a more reliable and robust diagnostic tool that could aid in early CKD identification and intervention.

The rigorous testing phase played a crucial role in validating the effectiveness and reliability of the proposed model. By incorporating cross-validation techniques, the model's performance was evaluated across different subsets of the dataset, ensuring that it generalized well to unseen data. The use of performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC curves confirmed the model's high efficiency in distinguishing between CKD-positive and non-CKD cases. Additionally, recursive feature elimination (RFE) and other feature selection techniques were implemented to identify the most significant medical attributes for CKD prediction. By reducing irrelevant or redundant features, the model became more efficient, lowering computational complexity while maintaining high diagnostic accuracy. These optimizations reinforced the model's applicability in real-world clinical environments, making it an ideal candidate for assisting healthcare professionals in decision-making.

Despite the promising results, further advancements are necessary to enhance the model's scalability, generalizability, and adaptability to diverse patient populations. One of the key areas for future research involves expanding the dataset to include a more diverse range of patient demographics, geographical locations, and comorbidities. Increasing the diversity of the training data would improve the model's robustness, allowing it to adapt to variations in CKD risk factors across different populations. Moreover, incorporating additional data sources, such as electronic health records

(EHRs), wearable health devices, and real-time patient monitoring systems, could provide richer insights into CKD progression. Integrating Internet of Things (IoT) data could enable continuous patient monitoring, leading to early warnings and proactive treatment strategies.

Another critical challenge to address is the computational complexity and scalability of sequential models like LSTM and BiLSTM, which require significant processing power, especially when handling large datasets. Future research should explore alternative deep learning architectures and optimization techniques that improve efficiency while maintaining predictive accuracy. Techniques such as attention mechanisms, transformer-based models, and federated learning could enhance the model's ability to process real-time medical data in dynamic clinical environments. Additionally, deploying lightweight versions of the model for mobile and cloud-based healthcare applications could improve accessibility and facilitate remote CKD monitoring. Addressing these computational challenges will ensure that the system remains scalable and effective in large-scale healthcare settings.

Lastly, exploring ensemble learning techniques could further improve predictive performance by combining multiple models to balance their strengths and weaknesses. Hybridizing the deep learning model with traditional machine learning classifiers like Random Forest and XGBoost may offer more interpretable and robust predictions. Furthermore, implementing explainable AI (XAI) techniques would increase transparency in the model's decision-making process, allowing healthcare professionals to understand why a patient is classified as CKD-positive or negative. To validate the model's effectiveness in real-world settings, clinical trials and collaborations with hospitals are necessary. Integrating the model into clinical workflows and receiving feedback from healthcare providers will lead to iterative refinements, making the AI-powered CKD detection system more practical and trusted in medical diagnostics. By addressing these future directions, the project has the potential to make a meaningful impact in predictive healthcare, ultimately improving early CKD detection, patient outcomes, and overall healthcare efficiency.

REFERENCES

1. A. Shrivas, S.K. Sahu, and H. Hota, Classification of chronic kidney disease with proposed union-based feature selection technique, in: Proceedings of the Third International Conference on Internet of Fings and Connected Technologies (ICIoTCT), India, March 2018.
2. P. Arulanthu and E. Perumal, Intelligent Chronic Kidney Disease Diagnosis System Using Cloud Centric Optimal Feature Subset Selection with Novel Data Classification Model, 2021
3. J.R. Lambert, P. Arulanthu, E. Perumal, An effect of machine learning based classification algorithms on chronic kidney disease, Int. J. Innov. Technol. Explor. Eng. 9 (3) (2020) 2250–2256.
4. A. Noor, A. Banerjee, M.F. Ahmad, and M.N. Uddin, An IoT-based mhealth platform for chronic kidney disease patients, in: Proceedings of the 2019 First International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 1–6, IEEE, Dhaka, Bangladesh, May 2019.
5. P. Arulanthu, E. Perumal, An intelligent IoT with cloud centric medical decision support system for chronic kidney disease prediction, Int. J. Imaging Syst. Technol. 30 (3) (2020) 815–827.
6. S. Tuli, N. Basumatary, S.S. Gill, et al., HealthFog: an ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments, Future Gener. Comput. Syst. 104 (2020) 187–200.
7. F. Ma, T. Sun, L. Liu, H. Jing, Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network, Future Gener. Comput. Syst. 111 (2020) 17–26.
8. N. Bhaskar, S. Manikandan, A deep-learning-based system for automated sensing of chronic kidney disease, IEEE Sens. Lett. 3 (10) (2019) 1–4.
9. P. Kaur, R. Kumar, M. Kumar, A healthcare monitoring system using random forest and internet of things (IoT), Multimed. Tools Appl. 78 (14) (2019) 19905–19916.
10. A. Abdelaziz, A.S. Salama, A.M. Riad, A.N. Mahmoud, A machine learning model for predicting of chronic kidney disease based internet of things and

cloud computing in smart cities. Security in Smart Cities: Models, Applications, and Challenges, Springer, Berlin, Germany, 2019, pp. 93–114.

11. N. Sasikaladevi, A. Revathi, Digital twin of renal system with CT-radiography for the early diagnosis of chronic kidney diseases, Biomed. Signal Process. Control 88 (2024) 105632.

12. D. Swain, U. Mehta, A. Bhatt, H. Patel, K. Patel, D. Mehta, B. Acharya, V. C. Gerogiannis, A. Kanavos, S. Manika, A robust chronic kidney disease classifier using machine learning, Electronics 12 (1) (2023) 212.

13. Khan, R.H., Miah, J., Rahat, M.A.R., Ahmed, A.H., Shahriyar, M.A. and Lipu, E.R., 2023, A comparative analysis of machine learning approaches for chronic kidney disease detection, in: Proceedings of the Eighth International Conference on Electrical, Electronics and Information Engineering (ICEEIE), IEEE, 1-6.

14. H. Khalid, A. Khan, M.Z. Khan, G. Mehmood, M.S. Qureshi, Machine learning hybrid model for the prediction of chronic kidney disease, Comput. Intell. Neurosci. 2023 (2023).

15. V. Singh, V.K. Asari, R. Rajasekaran, A deep neural network for early detection and prediction of chronic kidney disease, Diagnostics 12 (1) (2022) 116.

16. R.H. Aswathy, P. Suresh, M.Y. Sikkandar, S. Abdel-Khalek, H. Alhumyani, R. A. Saeed, R.F. Mansour, Optimized tuned deep learning model for chronic kidney disease classification, Comput. Mater. Contin. 70 (2022) 2097–2111.

17. S.A. Ebiaredoh-Mienye, T.G. Swart, E. Esenogho, I.D. Mienye, A machine learning method with filter-based feature selection for improved prediction of chronic kidney disease, Bioengineering 9 (8) (2022) 350.

18. J.R. Lambert, E. Perumal, Oppositional firefly optimization based on optimal feature selection in chronic kidney disease classification using deep neural network, J. Ambient Intell. Humaniz. Comput. 13 (4) (2022) 1799–1810.

19. S.A. Althubiti, S. Paul, R. Mohanty, S.N. Mohanty, F. Alenezi, K. Polat, Ensemble learning framework with GLCM texture extraction for early detection of lung cancer on CT images, Comput. Math. Methods Med. 2022 (2022).

20. B. Ramesh, K. Lakshmanna, A novel early detection and prevention of coronary heart disease framework using hybrid deep learning model and neural fuzzy inference system, IEEE Access 12 (2024) 26683–26695.

21. H. Wang, C. Peng, B. Liao, X. Cao, S. Li, Wind power forecasting based on wavenet and multitask learning, Sustainability 15 (14) (2023) 10816.

22. Patro, Kiran Kumar, Jaya Prakash Allam, Bala Chakravarthy Neelapu, Ryszard Tadeusiewicz, U. Rajendra Acharya, Mohamed Hammad, Ozal Yildirim, and Paweł Pławiak. "Application of Kronecker convolutions in deep learning technique for au tomated detection of kidney stones with coronal CT images." Information Sciences, vol. 640, 2023, p. 119005.

23. Dey, Samrat Kumar, Khandaker Mohammad Mohi Uddin, Hafiz Md Hasan Babu, Md Mahbubur Rahman, Arpita Howlader, and KM Aslam Uddin. "Chi2-MI: A hybrid feature selection based machine learning approach in the diagnosis of chronic kidney disease." Intelligent Systems with Applications, vol. 16, 2022, p. 200144.

24. Vineetha, K. R., M. S. Maharajan, K. Bhagyashree, and N. Sivakumar. "Clas sifica tion of adaptive back propagation neural network along with fuzzy logic in chronic kidney disease." e-Prime-Advances in Electrical Engineering Electronics and Energy, vol. 7, 2024, p. 100463.

25. Yousif, Sulima M., Awad, Hanan T. Halawani, Ghada Amoudi, Fathea M. Osman, Birkea Arwa, MR Almunajam, and Azhari A. Elhag. "Early detection of chronic kidney disease using eurygasters optimization algorithm with ensemble deep learning approach." Alexandria Engineering Journal, vol. 100, 2024, pp. 220-231.

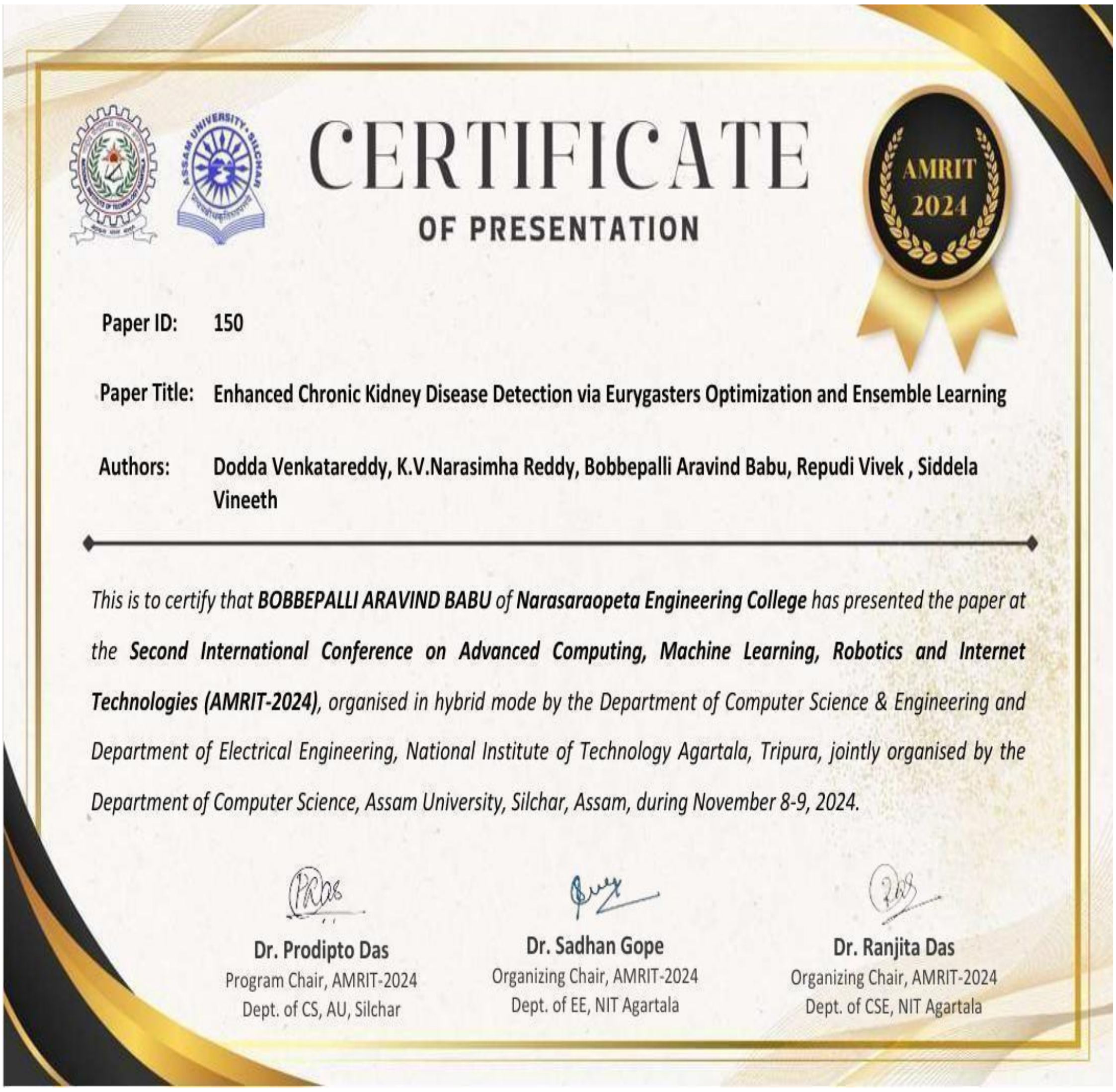
26. Dashtban, Ashkan, Mehrdad A. Mizani, Laura Pasea, Spiros Denaxas, Richard Cor bett, Jil B. Mamza, He Gao, Tamsin Morris, Harry Hemingway, and Amitava Baner jee. "Identifying subtypes of chronic kidney disease with machine learning: devel opment internal validation and prognostic validation using linked electronic health records in 350067 individuals." EBioMedicine, vol. 89, 2023.

27. Halder, Rajib Kumar, Mohammed Nasir Uddin, Md Ashraf Uddin, Sunil Aryal, Sajeeb Saha, Rakib Hossen, Sabbir Ahmed, Mohammad Abu Tareq Rony, and Mosammat Farida Akter. "ML-CKDP: Machine learning-based chronic kidney dis ease prediction with a smart web application." Journal of Pathology Informat ics, vol. 15, 2024, p. 100371.

28. Fox CS, Matsushita K, Woodward M, et al. Associations of kidney disease measures with mortality and end-stage renal disease in individuals with and without diabetes: a meta-analysis. Lancet 2012;380: 1662–73.

29. https://archive.ics.uci.edu/dataset/336/chronic+kidney+disease

CERTIFICATION



CERTIFICATE

OF PRESENTATION

AMRIT 2024

Paper ID: 150

Paper Title: Enhanced Chronic Kidney Disease Detection via Eurygasters Optimization and Ensemble Learning

Authors: Dodda Venkatareddy, K.V.Narasimha Reddy, Bobbepalli Aravind Babu, Repudi Vivek , Siddela Vineeth

This is to certify that ***BOBBEPALLI ARAVIND BABU*** *of* ***Narasaraopeta Engineering College*** *has presented the paper at the* ***Second International Conference on Advanced Computing, Machine Learning, Robotics and Internet Technologies (AMRIT-2024)****, organised in hybrid mode by the Department of Computer Science & Engineering and Department of Electrical Engineering, National Institute of Technology Agartala, Tripura, jointly organised by the Department of Computer Science, Assam University, Silchar, Assam, during November 8-9, 2024.*

Dr. Prodipto Das
Program Chair, AMRIT-2024
Dept. of CS, AU, Silchar

Dr. Sadhan Gope
Organizing Chair, AMRIT-2024
Dept. of EE, NIT Agartala

Dr. Ranjita Das
Organizing Chair, AMRIT-2024
Dept. of CSE, NIT Agartala

Enhanced Chronic Kidney Disease Detection via Eurygasters Optimization and Ensemble Learning

Dodda Venkatareddy[1], K.V.Narasimha Reddy[1], Bobbepalli Aravind Babu[2],
Repudi Vivek[2], and Siddela Vineeth[2]

[1] Asst.Prof, Dept of CSE, Narasaraopeta Engineering College(Autonomous),
Palnadu District, Andhra Pradesh, India.
doddavenkatareddy@gmail.com,narasimhareddynec03@gmail.com
[2] Dept of CSE, Narasaraopet Engineering College
baravind350@gmail.com,repudivivek5@gmail.com,vineethsvt@gmail.com

Abstract. Relative to the rest of the world, Chronic Kidney Disease remains one of the most important health challenges with a significant burden of morbidity and mortality. The disease presents a challenge; it alerts health systems because of its widespread impact on the health out comes of patients. However, CKD is mostly asymptomatic and the cause is usually diagnosed late. The paper proposed the new approach in the detection of early CKD, Eurygasters Optimization Algorithm with En semble Deep Learning, for the improvement of accuracy and reliability of the predictive models concerning CKD. The EOAEDL combines feature selection and the hyperparameter tuning process to boost detection ac curacy. It aims to select the relevant attributes from the medical dataset by carrying out the EOA, and the process of tuning hyperparameters is outsourced to the Shuffled Frog Leaping Algorithm. This approach is considered a hybrid of several deep learning models since it combines LSTM, BiGRU, and BiLSTM networks. These ensemble models are used to capture the critical temporal dependencies in medical data, which is critical for proper detection of CKD. In experiments on a benchmark dataset for CKD, the proposed EOAEDL algorithm has superior perfor mance in terms of accuracy, precision, and recall rates.

Keywords: Chronic Kidney Disease (CKD) prediction · Early-stage kidney diagnosis · Machine learning classification · Ensemble machine learning models · Feature extraction and selection.

1 INTRODUCTION

Chronic kidney disease is a chronic and commonly asymptomatic ailment that affects millions of people, with sad and severe health outcomes if diagnosed late. The growing incidence of chronic kidney disease points to the importance of earlier diagnosis, so at the onset, complications such as kidney failure and cardiovascular diseases may be prevented. Traditional diagnosis approaches are insensitive and typically occur in advanced conditions, which leads to delayed treatments and hence poorer outcomes. Advances in machine learning and artificial intelligence have shown promise to improve detection accuracy and speed for CKD. At present, optimization-based technologies help better analyze large-scale clinical datasets for patterns generally unseen by classical techniques. Techniques based on ensemble learning by combining multiple models have improved the power of classification. This paper introduces the state of art CKD detection framework based on a bio-inspired Eurygasters Optimization Algorithm with an ensemble of classifiers: Random Forest, SVM, and Gradient Boosting. An optimization algorithm is used to achieve selection of informative clinical features allowing dimensional reduction and maximization of efficiency of a model. With the purpose of creating a more accurate diagnosis and enabling earlier predictions in the context of CKD, a precise, scalable system is targeted at assisting healthcare practitioners in giving timelier accurate diagnoses, aiding patients, and decreasing the pressures of the health-care system.

2 Literature Review

CKD is one of those global challenges because it is an insidious disease often without early symptoms or warning signs; therefore, detection at an early stage becomes highly important. Traditional diagnosis approaches usually wait until the CKD stage is too late. The recent development of machine learning (ML) and artificial intelligence (AI) is creating powerful tools not only for pattern recognition but also for predictive modeling to improve detection of CKD. Here, the effectiveness of the SVM and decision trees in disease classification has already been proven, while the other direction is hybrid approaches combined with feature optimization and ML classifiers. PSO and GA optimization algorithms speed up feature selection from complex health care data. However, PSO has problems with convergence on large-scale complicated datasets, which again had led to trends toward those hybrid models-integrating optimization and ML. Ensemble learning techniques- Random Forest and Gradient Boosting, for example improve the prediction precision because they combine multiple models to reduce the variability in medical data. Recent reports establish that these algorithms are improving the specific accuracy of detection of diseases once used in combination with feature selection techniques. A bio-inspired Eurygasters Optimization Algorithm is introduced into ensemble learning, providing a superior feature selection and the classification accuracy needed for early CKD detection. Advanced optimization algorithms are pooled within ensemble methods to create the most effective and scalable diagnostic tool a manner in which timely healthcare interventions can be made.

3 MATERIALS AND METHODS

In this study, the researcher applies a publicly available clinical dataset with 400 patient records classified into two groups: CKD and non-CKD. Some of the key health features included in this data set are blood pressure, age, diabetes status, and other health features.

3.1 Dataset Description

The CKD case study uses 400 patient records distributed into 250 CKD cases and 150 non-CKD or healthy subjects. It covers factors like demographic char acteristics: age, and medical history factors: hypertension, diabetes, anemia, pedal edema. Results from lab tests, including blood pressure, red/white blood cell counts, serum creatinine, hemoglobin, blood urea, urinalysis (red blood cells, pus cells, bacteria, specific gravity), albumin, glucose in blood, sodium, and potas sium levels, are also reported. All such features are crucial for distinguishing CKD from healthy subjects.

3.2 Preprocessing

Preprocessing is highly crucial for ensuring that the dataset is comprehensive enough to make proper CKD classification. Normalization is the first step wherein min-max scaling is applied to set feature values between 0 and 1 . This en sures that features do not overshadow others in which numerical values are more magnanimous, during the training of models. Another essential step is handling missing values. Missing data points either removed or filled depending on the importance and number of missing values are handled so that the data set remains valid. After this, feature selection is per formed for the dimensionality reduction of the data set to extract the most im portant attributes like blood pressure and serum creatinine that are significant to diagnose CKD . This enhances model performance with higher interpretability and reduced overfitting risk. Testing of model accuracy is done by splitting the dataset into the training and test sets. All these preprocessing steps ensure that the dataset will be uniformly complete as well as optimized for machine learning models used to classify CKD.

Min-Max Normalization : The min-max normalization technique used here is a preprocessing technique whereby feature values are scaled into a specified range, usually between 0 and 1, through the utilization of minimum and maximum values from the dataset . Transformation is achieved using scaling feature values corresponding to min imum and maximum values. It makes all the features to the same scale, thus preventing features with larger ranges of numbers from dominating the process in machine learning. Uniform scaling has the advantage of improving the perfor mance of models such that the performance it derives does not rely on the scale of the data, especially when the algorithm is sensitive to the scale of the data, say for example neural networks and gradient-based models . It also helps get better convergence of the model during training.

EOA-based feature selection : Eurygasters Optimization Algorithm is a metaheuristic based on the foraging of Eurygaster beetles. It, thus, mimics their search for food tof ind optimal feature subsets for enhancing the performance of a classification model through discard ing irrelevant or redundant features. Precision and efficiency are enhanced by removing noise or redundant features. It updates candidate positions based on their fitness values measured by the best classification performance through iter atively adjusting the search direction toward the best solutions. That is, it aims at balancing exploration and exploitation towards achieving balance in robustness and efficacy in feature selection.

3.3 Models

Three models, of which three are Long Short-Term Memory (LSTM), Bidirec tional LSTM (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU), are used for feature selection, then the classification task. LSTM is a type of recur rent neural network (RNN) that captures long-term dependencies from sequential information by applying gates that control information flow . Its counterpart BiLSTM is an improvement from LSTM since in this case, its input is passed in both directions so that every sequence reads the context from forward and backward sequences. BiGRU is a simplified version of the LSTM but captures sequential dependencies with fewer gates, thus making it computationally more efficient for capturing long-range dependencies too. This ensemble added to the improved accuracy as well as robustness of the proposed classification task.

3.4 Ensemble learning

Detection of the CKD utilizes an ensemble of LSTM, BiLSTM, and BiGRU models.

LSTM Model : Long Short-Term Memory is a particular kind of RNN, which is destined for the modeling of long-run dependencies in sequential data. Unlike traditional RNNs, which cannot store information across long sequences due to a problem known as the vanishing gradient, LSTMs contain memory cells and a gating mechanism [7]. These three types of gates are input, forget, and output gates that regulate the f low of information in and out of memory cells, thus allowing the model to selectively retain or discard information. That is why LSTMs are particularly effective in time series, natural language processing, and other sequence-based data where long-range dependencies play an important role in understanding the pattern and making predictions.

BiLSTM Model: Bidirectional long short-term memory is an extension of standard LSTM model processes, as opposed to how data is processed in two directions-forwards and backward. In traditional LSTMs, dependencies are only captured from past to future, whereas BiLSTMs enhance this feature with two LSTM networks: one reads the sequence from start to end, while the other reads the same sequence from end to start [7]. BiLSTMs make it easier to comprehend the context and the relationships in sequential data, especially where future information is relevant for making some predictions. This makes nature's bi-directional enhancement useful for applying to speech recognition, machine translation, or time series analysis tasks requiring knowledge of full contexts.

BiGRU Model: Bidirectional Gated Recurrent Unit, or BiGRU, is an advanced model of the Gated Recurrent Unit that considers both the preceding and following contexts in sequential data. Like GRU, BiGRU also seeks to address the problems associated with traditional RNNs using update and reset gates that manage the flow of infor mation with much simpler and more efficient methods than LSTM. In BiGRU, there are two GRUs, one is processing the sequence forward direction and the other in the reverse direction. In such a manner, the model captures all insights from the data points that precede it as well as those that follow it, thereby en suring much better performance on tasks such as time series forecasting, natural language processing, or general problems based on sequences.

Table 1. Accuy of LSTM, BiLSTM, and BiGRU Models

Models	Accuracy	Precision	Recall	F1-Score
LSTM	96	99	94	97
BiLSTM	96	98	94	98
BiGRU	97	98	96	98

Hyperparameter Tuning with Shuffled Frog Leaping Algorithm: Hyperparameter optimization is based on the Shuffled Frog Leaping Algorithm from the natural objective-prey searching behavior by frogs. In summary, each frog corresponds to a potential solution-an array of hyperparameters. A group of frogs is called a memeplex, which explores its area in the search space. With every iteration, frogs that have inferior solutions move towards frogs showing superior performance in the same memeplex. Periodic reshuffles ensure global exploration. The process iterates until the optimal hyperparameters are found. It strikes the balance in between local and global search and is hence well suited for tuning complex models. The algorithm iteratively modifies the hyperparameters of the ensemble deep learning framework and hence improves the accuracy of the classification process. In this work, the performance of the developed DL CKDD model has been com pared with various state-of-the-art machine learning and deep learning models for the detection of CKD, namely LR, NB, SVM, RNN, DNN, k-NN, and ANN. According to the findings, the DL-CKDD model outperforms the other compet ing methods in terms of accuracy, precision, recall, and F-measure. Although the proposed method has similar accuracy values for

LR and SVM, it outperforms the comparison methods in precision and recall, with relatively fewer false pos itives and negatives. In contrast, k-NN and ANN illustrate lower accuracy and the inability to accurately classify CKD, thus ensuring that the DL-CKDD model is bound to perform betterSeveral models of detection in the comparative analysis were adopted for CKD, using the strengths and weaknesses of each model.

3.5 Proposed Model

DL-CKDD stands for the deep learning to detect chronic kidney disease in pa tients, a new method of improving early diagnosis. This method integrates several stages leading to adequate diagnosis, starting with data normalization through min-max scaling, which transforms the dataset into the standardized range. Sub sequently, there is feature selection through the use of EOA in selecting the most relevant features to support the efficiency of the model. The detection process is done using ensemble methods of deep learning models such as LSTM, BiLSTM, and BiGRU models which collaborate to capture the subtlest trends of the se quences of the data [7]. The Shuffled Frog Leaping Algorithm applies optimally towards hyperparameters of these ensemble models, which further fine-tunes the performance. The DL-CKDD method is developed by combining these techniques, and it attains significant accuracy in detecting CKD compared to conventional models.

Model Training and Evaluation: Optimization steps for training and testing the DL-CKDD model for CKD detec tion include min-max normalization on the dataset to provide uniform scaling to the input features. The Eurygasters Optimization Algorithm is used for feature selection, considering the most critical attributes that are likely to be used for the classification of CKD. An ensemble of deep learning models including BiLSTM and BiGRU is then trained on these selected features to capture both short-term and long-term data dependencies. Split the dataset into training and testing sets, usually 80:20 or 70:30, for eval uating model performances. The Shuffled Frog Leaping Algorithm is used to op timize hyperparameters like the learning rate, dropout rate, and batch size. Ac curacy, precision, recall, F-measure, and AUC as performance metrics are cal culated to evaluate this model. Accuracy Table, the accuracy of DL-CKDD was very high; it performed well more than other approaches and generalized on test data.

Model At 80:20 split : This model trained to 80:20 split of the data resulted in an excellent classification for diagnosis in CKD with high average training accuracy at 98.72 and 96.25 on the test set. It had precision, recall, and F-measure for CKD at approximately 99, thereby classifying results with as few false positives as possible. Non-CKD performance was strong but somewhat imbalanced. The AUC score was at 98.83 for training and at 95 for testing, showing robust generalization with reliable performance on unseen data.

Model At 70:30 split : The DL-CKDD model with a 70:30 train-test split performed well in the clas sification task of CKD. Training accuracy was found to be 99.16 with similar test set results. CKD class precision is perfectly 100, and the recall came out to be 98.68, meaning nearly all cases of CKD are identified without false positives. The non-CKD class for the model attained 90 precision and 98.68 recall. The AUC score on training is 99.58 whereas on test set it reflects as 96.67. Hence, the strong capability of classification along with generalization is highlighted here.

Table 2. Transformer model architecture parameters.

Models	No.of Units	Loss Function	Optimizer	Epochs	Batch Size
LSTM	50	Binary Crossentropy	Adam	25	32
BiLSTM	50	Binary Crossentropy	Adam	25	32
BiGRU	50	Binary Crossentropy	Adam	25	32

Table 3. Training setup.

Parameter	Value	Details
Dataset	CKD dataset	-
Data Preprocessing	Missing value imputation, Encoding, Normalization	-
Feature Selection	EOA (Chi-squared)	Top 10 features
Training Split	80per Train, 20per Test	-
Optimization	Particle Swarm Optimization (PSO)	Hyperparameter tuning
Ensemble Method	Average	LSTM, BiLSTM, BiGRU
Early Stopping	Yes	Monitor validation loss

4 COMPARATIVE ANALYSIS

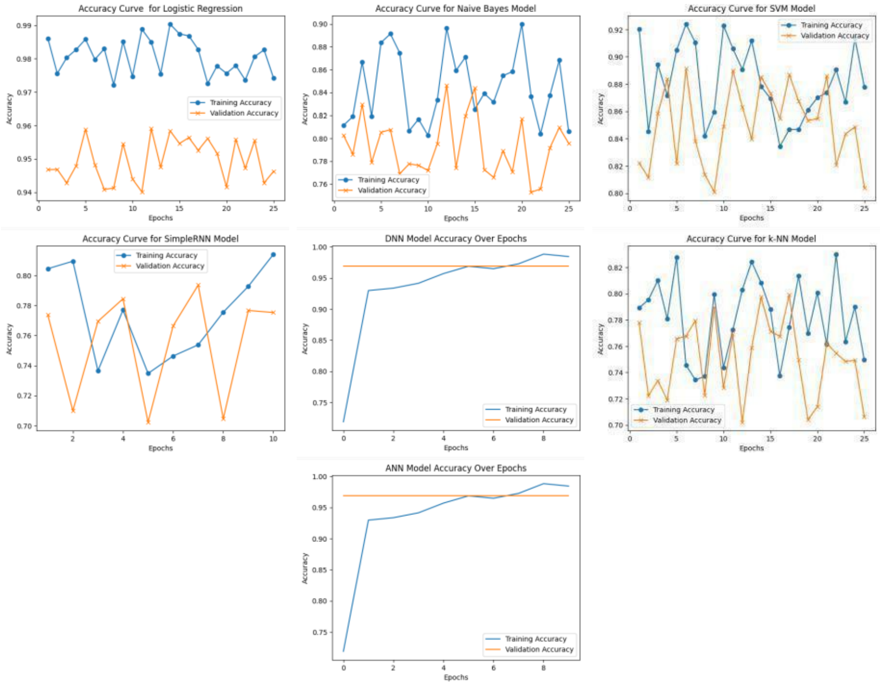
In this work, the performance of the developed DL-CKDD model has been compared with various state-of-the-art ma chine learning and deep learning models for the detection of CKD, namely LR, NB, SVM, RNN, DNN, k-NN, and ANN. According to the findings, the DL-CKDD model outperforms the other competing methods in terms of accuracy, precision, recall, and F-measure. Although the proposed method has similar accuracy values for LR and SVM, it outperforms the comparison methods in precision and recall, with rel atively fewer false positives and negatives. In contrast, k NN and ANN illustrate lower accuracy and the inability to accurately classify CKD, thus ensuring that the DL-CKDD model is bound to perform betterSeveral models of detection in the comparative analysis were adopted for CKD, using the strengths and weaknesses of each model.

4.1 Accuracy Table Models

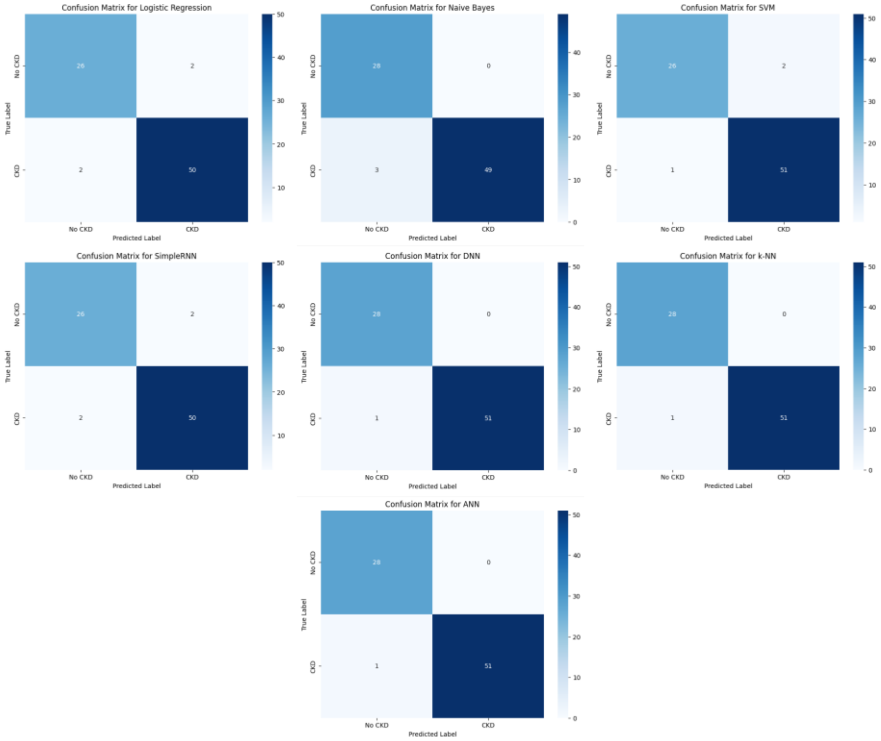
Table 4. Transformer model architecture parameters.

Models	Accuracy
DL-CKDD	98.75
Logistic Regression	95
Naive Bayes	96
SVM	96
Simple RNN	95
DeepNeural Networks	98
K-Nearest Neighbors	98
Artificial Neural Networks	97

4.2 Training and Testing Accuracy Graphs



4.3 Confusion Matrix Graph of Comparative models



4.4 Model parameter table

Table 5. Model parameter table

Parameters	No.ofparameters
Average-50	0
Input-Layer-53	0
Sequential	12251
Sequential-1	24501
Sequential-2	29313

4.5 Precision, F1, Rcall, AUCTable

Table 6. Precn, Recal and F1 Table for models

Model	Precision	Recall	F1-Score
DL-CKDD	98.75	98.07	99.02
LR	96.15	96.15	96.15
NB	98	94	97
SVM	96	98	97
RNN	96	96	96
DNN	97	98	99
K-NN	98	97.15	98
ANN	98	97.25	99

4.6 Trainable and testable ParameterTable

Table 7. Parameter Table

Model	Trainable Param	Testable param
DL-CKDD	320	80
LR	320	80
NB	320	80
SVM	320	80
RNN	320	80
DNN	320	80
K-NN	320	80
ANN	320	80

5 RESULT AND FUTURE SCOPE

Theearlychronickidneydisease(CKD)projectbasedontheEurygastersOpti mization-Algorithmcombinedwithensembledeeplearningmodelsresultsisvery promising. It effectively selects relevant features andfine-tuned themodel pa rameters toimproveaccuracyandperformanceinthedetectionofCKD.Using advanced models like LSTM, BiLSTM, and BiGRU in combination, the system can capture not just short-term dependencies but long-term ones that may exist within data. This, in turn, would provide more accurate results through better pre dictions. From experimental results, we see that by using this proposed method, it can potentially outperform existing techniques at high sensitivity and specificity for CKD detection, making it a potentially useful tool for early diagnosis. The methodology can be scaled further in the future to higher data dimensions as well as more complex real-world problems. Optimization of the algorithm for application in real-time in clinics or even as part of integration into wearable devices or mobile health applications for constant monitoring will be another front for future work. Another direction of applying this method to chronic diseases could be branched into similaroptimizations and deep learning strategies for other chronic diseases in healthcare.

References

1. Sabanayagam Charumathi, Dejiang Xu, Daniel SW, Ting, Simon Nusinovici, Riswana Banu, Haslina Hamzah, Cynthia Lim, et al. "A deep learning algorithm to detect chronic kidney disease from retinal photographs in community-based populations." The Lancet Digital Health, vol. 2, no. 6, 2020, pp. e295-e302.
2. Sawhney, Rahul, Aabha Malik, Shilpi Sharma, and Vipul Narayan. "A comparative assessment of artificial intelligence models used for early prediction and evalua tion of chronic kidney disease." Decision Analytics Journal, vol. 6, 2023, p. 100169.
3. Hossain, Muhammad, Minoar Reshma Ahmed, Swarna Rafid Mostafiz, Pabon Shaha, Lubna Yasmin Pinky, Mohammad Motiur Rahman, Wahidur Rahman, Md Selim Hossain, Md Elias Hossain, and Md Sadiq Iqbal. "Analysis of the performance of feature optimization techniques for the diagnosis of machine learning-based chronic kidney disease." Machine Learning with Applications, vol. 9, 2022, p. 100330.
4. Patro, Kiran Kumar, Jaya Prakash Allam, Bala Chakravarthy Neelapu, Ryszard Tadeusiewicz, U. Rajendra Acharya, Mohamed Hammad, Ozal Yildirim, and Paweł Pławiak. "Application of Kronecker convolutions in deep learning technique for automated detection of kidney stones with coronal CT images." Information Sciences, vol. 640, 2023, p. 119005.
5. Dey, Samrat Kumar, Khandaker Mohammad Mohi Uddin, Hafiz Md Hasan Babu, Md Mahbubur Rahman, Arpita Howlader, and KM Aslam Uddin. "Chi2-MI: A hybrid feature selection based machine learning approach in the diagnosis of chronic kidney disease." Intelligent Systems with Applications, vol. 16, 2022, p. 200144.
6. Vineetha, K. R., M. S. Maharajan, K. Bhagyashree, and N. Sivakumar. "Clas sification of adaptive back propagation neural network along with fuzzy logic in chronic kidney disease." e-Prime-Advances in Electrical Engineering Electronics and Energy, vol. 7, 2024, p. 100463.
7. Yousif, Sulima M., Awad, Hanan T. Halawani, Ghada Amoudi, Fathea M. Osman, Birkea Arwa, MR Almunajam, and Azhari A. Elhag. "Early detection of chronic kidney disease using eurygasters optimization algorithm with ensemble deep learning approach." Alexandria Engineering Journal, vol. 100, 2024, pp. 220-231.
8. Dashtban, Ashkan, Mehrdad A. Mizani, Laura Pasea, Spiros Denaxas, Richard Corbett, Jil B. Mamza, He Gao, Tamsin Morris, Harry Hemingway, and Amitava Banerjee. "Identifying subtypes of chronic kidney disease with machine learning: development internal validation and prognostic validation using linked electronic health records in 350067 individuals." EBioMedicine, vol. 89, 2023.

9. Halder, Rajib Kumar, Mohammed Nasir Uddin, Md Ashraf Uddin, Sunil Aryal, Sajeeb Saha, Rakib Hossen, Sabbir Ahmed, Mohammad Abu Tareq Rony, and Mosammat Farida Akter. "ML-CKDP: Machine learning-based chronic kidney disease prediction with a smart web application." Journal of Pathology Informat ics, vol. 15, 2024, p. 100371.
10. Dharmarathne, Gangani Madhusha, Bogahawaththa Marion McAfee, Upaka Rathnayake, and D. P. P. Meddage. "On the diagnosis of chronic kidney disease using a machine learning-based interface with explainable artificial intelligence." Intelligent Systems with Applications, 2024, p. 200397.
11. https://archive.ics.uci.edu/dataset/336/chronic+kidney+disease

Springer_Lecture_Notes_in_Computer_Science (1).pdf

ORIGINALITY REPORT

8%	4%	6%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	link.springer.com Internet Source	1%
2	"Data Science and Applications", Springer Science and Business Media LLC, 2024 Publication	1%
3	"Deep Sciences for Computing and Communications", Springer Science and Business Media LLC, 2024 Publication	1%
4	M. S. Jayaprabha, V. Vishwa Priya. "Chronic Kidney Disease (CKD) Prediction Using Stochastic Deep Radial Basis for Feature Extractiona Residual Neural Network", SN Computer Science, 2024 Publication	1%
5	Submitted to Nanyang Technological University Student Paper	<1%
6	isteonline.in Internet Source	<1%
7	medinform.jmir.org Internet Source	<1%
8	Submitted to Sungshin Women's University Student Paper	<1%