International Journal of
Human Computations & Intelligence
Vol. 04, Issue. 06, 2025

RESEARCH ARTICLE OPEN ACCESS

Hybrid Ensemble – Deep Learning Framework for Chronic Kidney Disease Prediction

G Ramasubba Reddy[1] . A Jyothi[2] . G Amulya[2] . Saritha Dasari[2] . Akki Rajasekhar Reddy[3]

[1]Department of CSM, Sai Rajeswari Institute of Technology, Proddatur, India.
[2]Department of Computer Science and Engineering, G Narayanamma Institute of Technology and Science, Hyderabad, India.
[3]Department of Humanities and Science, Sai Rajeswari Institute of Technology, Proddatur, India.

DOI: **10.5281/zenodo.17191849**
Received: 12 September 2025 / Revised: 20 September 2025 / Accepted: 24 September 2025
©Milestone Research Publications, Part of CLOCKSS archiving

Abstract – Early detection of chronic kidney disease (CKD) is crucial, as delayed diagnosis can lead to irreversible damage and high mortality rates. Traditional diagnostic methods often struggle to capture the complex, non-linear interactions among diverse clinical indicators such as blood pressure, haemoglobin, blood glucose, and serum creatinine. To address this limitation, the present study evaluates the performance of a proposed CatBoost-DeepNet hybrid framework against several well-established machine learning classifiers. The framework integrates the feature selection and interpretability strengths of CatBoost with the representational capacity of a deep neural network, enabling more accurate and reliable CKD prediction. Based on a real-world dataset of 400 patient records and 25 clinical features, we compared CatBoost-DeepNet to nine traditional and ensemble models, i.e., K-Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting, AdaBoost, XGBoost, LightGBM, and Extra Trees. Results indicate that CatBoost-DeepNet consistently outperformed baselines with a test accuracy of 99.17%, precision of 99.50%, recall of 99.00%, and an F1 score of 99.25%. Confusion matrix assessment also confirmed the diagnostic reliability of the model with zero false negative and just a single false positive. These findings suggest that CatBoost-DeepNet is a strong, generalizable, and clinically valuable platform for early prediction of CKD.

Index Terms – Chronic Kidney Disease, CatBoost-DeepNet, Ensemble Learning Algorithms, Deep Learning, Machine Learning, Medical Data Mining, Clinical Decision Support.

MILESTONE
RESEARCH.IN
OPEN ACCESS

ISSN (Online): 2583-5696
Int. Jr. of Hum Comp. & Int.

I. INTRODUCTION

One of the most significant diseases of this era is chronic kidney disease (CKD), which persists in hundreds of thousands of people across the globe and is among the top diseases causing morbidity and mortality [1]. Chronic kidney disease (CKD) is not diagnosed easily and mostly goes undetected, so close diagnosis is highly difficult but necessary for improved patient outcomes and minimizing medical expenses [2]. The combined action of multiple risk factors is rarely given in the traditional diagnosis, depending on absolute medical criteria like creatinine, blood pressure, or proteinuria [3]. As a result, treatment is unavailable and expensive, and the majority of patients receive a diagnosis after their disease has been stabilized [4].

In addition to mere lab results, various interactive elements, such as lifestyle choices, demographics, and comorbidities, affect the CKD progression in actual life [5]. Depending on other conditions such as diabetes, hypertension, or coronary artery disease, two persons can have identical kidney function test results but may have different courses of disease [6]. These nonlinear interactions are impossible to quantify with standard statistical methods, which renders utilization of predictive risk estimates in clinical practice challenging. This is where machine learning (ML) and artificial intelligence (AI) have gained popularity. Healthcare is using machine learning (ML) techniques more and more for disease diagnosis, prognosis, and treatment optimization because of their capacity to uncover hidden patterns in complicated datasets [7]. For CKD, researchers have employed classifiers such as logistic regression, support vector machines (SVM), K-nearest neighbors (KNN), random forest, gradient boosting, and ensemble techniques [8–10]. Even though such models have shown promise, they normally experience either overfitting—whereby the model memorizes training data without generalizability—or underfitting, failing to learn important patterns [11].

To tackle such problems, this research presents a deep learning-driven Artificial Neural Network (ANN) for specifically predicting CKD. As opposed to traditional models, ANNs can learn complex, non-linear patterns between heterogeneous clinical features easily. We further incorporate dropout and early stopping methods into our model to make it stable and avoid overfitting. The novelty of this approach is that it is able to learn feature interactions—such as how blood pressure and hemoglobin collectively influence risk—without any need for human feature engineering. We contrasted our ANN with some popular classifiers, i.e., logistic regression, random forest, SVM, and KNN, on the standard UCI CKD dataset. Our experiments demonstrate that the ANN performed better than all the baselines across different evaluation metrics with higher accuracy, precision, recall, and F1-score. More importantly, the model showed superior generalization ability, justifying its strength as a reliable decision-support tool for doctors.

This paper makes three primary contributions:

- We propose a robust ANN-based system for CKD early prediction as a binary classification problem.
- We make a comprehensive comparison with traditional machine learning models to disclose their strengths and limitations.
- We provide a detailed performance analysis, e.g., generalization behavior, confusion matrix evaluation, and indications of underfitting or overfitting.

The result of this study can empower policymakers, practitioners, and healthcare organizations to adopt cutting-edge predictive models for CKD, with the outcome being earlier treatment, more personalized care, and reduced disease burden.

II. LITERATURE SURVEY

Computational intelligence and machine learning have also, in the past few years, been applied to improve early prediction of chronic kidney disease (CKD). Due to CKD being asymptomatic at the initial stages, early detection by predictive modeling has also been a significant research area, and the majority of studies have utilized classical algorithms, ensemble methods, and deep learning. Ghosh et al. [1] also did a comparative analysis with well-known classifiers such as Logistic Regression, Random Forest, AdaBoost, and XGBoost, and a recently proposed Hybrid Model. In their study, using the UCI CKD database, they showed that the Hybrid Model outperformed the rest consistently to an extent of nearly 95% accuracy, and established the benefits of integrating various algorithms.

Halder et al. [2] approached the problem differently by suggesting ML-CKDP, which not only enhanced data preprocessing with imputation, scaling, and feature selection through a large number of factors but also had an astute web application for on-demand prediction. They evaluated seven classifiers of which Random Forest and AdaBoost were the top performers, both of which achieved 100% accuracy in several validation methods, offering a high potential for clinical deployment. Interpretability was what Arif et al. [3] tackled, built an interpretable CKD prediction model using a multilayer perceptron with the incorporation of LIME. Their results showed that even if the MLP had very good predictive performance, its greatest contribution was transparency, to guide clinicians on why predictions were made, a step closer to solving the "black box" issue of ML in medicine.

In another line of work, Zhu et al. [4] concentrated on CKD patients at risk of cardiovascular disease, using LASSO regression for feature selection and building models with seven ML classifiers. They found Extreme Gradient Boosting (XGBoost) to be the most effective, achieving an AUC of 0.89, and identified key predictors such as age, hypertension, and sodium ion levels, highlighting the value of CKD-related comorbidities in prediction models. Some researchers have explored hybrid and neuro-fuzzy approaches. Praveen et al. [5] proposed a Neuro-Fuzzy model based on ML and image processing techniques for fibrosis detection in kidney tissues. Compared to conventional classifiers like SVM and KNN, their approach reached an accuracy of 97%, showing that integrating medical imaging with ML can improve early-stage diagnosis.

Vanathi et al. [6] examined multiple groups of algorithms, including ensemble trees, KNN, SVM, and ANN, reporting that their ANN-based approach achieved up to 99.2% accuracy for early CKD prediction. Their study confirmed that neural networks remain powerful in capturing complex, nonlinear patterns in medical data. A systematic review by Khalid et al. [7] reinforced these observations, showing that across 13 studies, AI/ML consistently achieved high sensitivity and specificity in identifying CKD progression, particularly when incorporating biomarkers and longitudinal data such as serum albumin and eGFR. Baswaraj et al. [8] explored both multiclass and binary classification methods for CKD stage prediction. Using Random Forest, SVM, and Decision Trees, they demonstrated that Recursive Feature Elimination (RFE) with cross-validation significantly improved predictive accuracy, with Random Forest outperforming the other models in their experiments.

Rahman et al. [9] stressed enhanced ensemble techniques and suggested a framework combining MICE imputation, SMOTE for imbalance handling, and recursive feature elimination. The highest performing model, LightGBM, with an accuracy of 99.75% and outperforms other state-of-the-art ensemble methods. Their results revealed that optimized feature selection and balancing methods play an important role in enhancing the performance of a classifier. Yamini et al. [10] employed six ML algorithms, i.e., Random Forest, CatBoost, and XGBoost, for the prediction of CKD. As can be seen in the experiments depicted, RF, CB, and XGB performed a consistent accuracy of around 95%, which was better than models like Gaussian Naive Bayes and Decision Trees. In this paper, the efficacy of gradient-boosting and ensemble models in real clinical prediction tasks was established.

Isaza-Ruget et al. [11] built time-to-event models for renal replacement therapy need prediction in patients with stage 3–5 CKD with excellent predictive accuracy of 0.89 by C-index. Similarly, Zheng et al. [12] compared Cox regression and Random Survival Forests and found that the former offered better nonlinear modeling, bested by XGBoost for classification. Patil and Choudhary [13] proposed a Cat Swarm Optimization-optimized hybrid classification model with neural networks and LSTM and demonstrated that the technique was better than traditional classifiers. Ekundayo [14] adopted a more general strategy, whose goal was to develop modeling of CKD development by Random Forests, Gradient Boosting, and SVM. Personalized treatment design and patient-individual risk stratification were among their top priorities. Data quality and interpretability were likewise mentioned to be key points under discussion, as well as how ML might possibly transform the treatment of patients if applied within health care systems.

Saif et al. [15] explored deep ensemble models employing CNN, LSTM, and BLSTM architectures. The model was 98% and 97% accurate, respectively, in predicting CKD risk 6–12 months ahead to highlight the strength of deep learning in preventive medicine. The articles confirm that although classical ML methods like SVM and Random Forest are still maintaining their relevance, deep learning and combinations are shattering predictive accuracy performance levels. Coupled with this, interpretability, feature engineering, and incorporation into real-time systems remain the key issues of concern for bringing prediction models to a point of reliability and deployability on an extended scale in medicine.

III. METHODS & MATERIALS

We utilized the chronic kidney disease (CKD) dataset in this research work with 400 patient records with 25 clinical parameters, such as blood pressure, blood sugar level, haemoglobin level, serum creatinine, and red and white blood cell count. The target label represents the class outcome: 0 for CKD and 1 for non-CKD. The preprocessing and feature engineering pipeline was designed to ensure robust experimental outcomes. All missing values were removed, and categorical variables were binary-encoded. Numerical features were normalized to a common scale to prevent bias from varying magnitudes. Feature selection was carried out to identify the most relevant predictors for classification. After preprocessing, multiple machine learning models were implemented, ranging from classical classifiers (KNN, Decision Tree, Random Forest) to ensemble learners (XGBoost, CatBoost, LightGBM, Extra Trees) and a deep learning-based ANN. Each model was trained and validated using an 80:20 data split. Finally, model performance was assessed using standard classification metrics—Accuracy, Precision, Recall, and F1-score. These metrics provided a comprehensive evaluation of the predictive capability of each algorithm. Figure 1 illustrates the overall research methodology followed in this work.

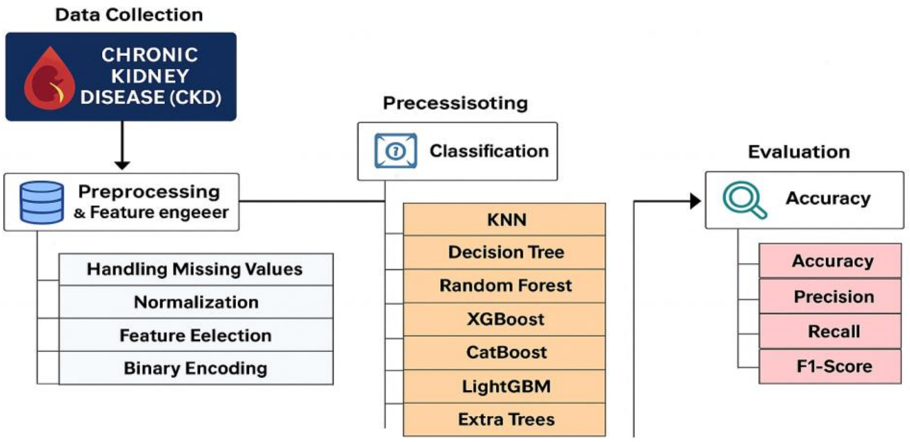

Fig. 1: Graphical representation of the overall research methodology

A. Dataset Description

In this study, we utilized a clinical dataset titled the Chronic Kidney Disease (CKD) Dataset, originally sourced from the UCI Machine Learning Repository. The dataset simulates real-world patient records collected over two months in India, encompassing a variety of clinical, hematological, and physiological measurements. It contains 400 anonymized patient entries, each described by 25 attributes related to kidney health, metabolic function, and general diagnostics. These features include quantitative variables such as blood glucose random levels, serum creatinine, hemoglobin levels, and blood pressure, as well as categorical medical indicators such as red blood cell condition, pus cell clumps, and appetite status. This combination of category and numerical variables makes the dataset ideal for real-world applications of machine learning involving both supervised classification and exploratory data analysis.

The target variable, classification, indicates whether the patient has or has not been clinically diagnosed with chronic kidney disease (CKD) or not (notckd) and frames the issue as a suitable binary classification problem for algorithms such as logistic regression, decision trees, and ensemble models. From a data quality perspective, the dataset contains missing values for most of the features. In the spirit of our research, rows with any missing values were dropped entirely to have a purer dataset for initial modeling and dimensionality reduction tasks. This conservative cleaning step is typical of standard pre-processing approaches in clinical data analysis, where missing records can predispose results to bias. A comprehensive description of all dataset features is shown in Table 1 below.

Table 1: Summarizing the Features of the Chronic Kidney Disease Dataset

Column Name	Description
id	Unique index for each patient record (range: 0 to 399)
age	Age of the patient in years (range: 2 to 90)
bp	Blood pressure in mm/Hg (range: 50 to 180)

sg	Specific gravity of urine (range: 1.005 to 1.025, indicates kidney concentration ability)
al	Albumin levels in urine (scale: 0 to 5)
su	Sugar levels in urine (scale: 0 to 5)
rbc	Red blood cell condition (categorical: normal/abnormal)
pc	Pus cell condition (categorical: normal/abnormal)
pcc	Pus cell clumps presence (categorical: present/notpresent)
ba	Bacteria presence (categorical: present/notpresent)
bgr	Blood glucose random test in mg/dl (range: 22 to 490)
bu	Blood urea in mg/dl (range: 1.5 to 391)
sc	Serum creatinine in mg/dl (range: 0.4 to 76)
sod	Sodium levels in mEq/L (range: 4.5 to 163)
pot	Potassium levels in mEq/L (range: 2.5 to 47)
hemo	Hemoglobin in g/dl (range: 3.1 to 17.8)
pcv	Packed cell volume (percentage; mostly numerical with some nulls)
wc	White blood cell count (in cells/cumm; contains noisy values and nulls)
rc	Red blood cell count (in millions/cumm; contains noisy values and nulls)
htn	Hypertension (boolean: true/false; contains many nulls)
dm	Diabetes mellitus (categorical: yes/no)
cad	Coronary artery disease (categorical: yes/no)
appet	Appetite (categorical: good/poor)
pe	Pedal edema (boolean: true/false; contains nulls)
ane	Anemia (boolean: true/false; contains nulls)
classification	Target variable: chronic kidney disease (ckd) or not (notckd)

B. Data Preprocessing and Feature Engineering

For offering effective learning and unbiased disease classification, an efficient preprocessing pipeline was developed on the Chronic Kidney Disease (CKD) dataset. The dataset is a combination of numerical and categorical variables for various clinical, metabolic, and diagnostic measurements. The original dataset consisted of 400 instances and 25 features, including continuous laboratory results (e.g., blood urea, serum creatinine) and categorical indicators (e.g., red blood cell status, presence of bacteria, appetite status). The target variable class represents the binary classification outcome: CKD (0) or Not CKD (1).

1. Handling Missing Data: An initial inspection revealed a considerable number of missing values, particularly in clinical fields like red/white blood cell counts, packed cell volume, sodium, and potassium. These were addressed using a dual strategy:
   For numerical features, we used random sampling imputation to preserve variance. For any numerical feature x_i with missing entries, a set of non-null values was randomly sampled and injected at the indices of missing values:

$$\mathrm{x}_i^{null} = \mathrm{RandomSample}\ (\mathrm{x}_i^{non-null})$$

   For categorical features, those with few missing values (e.g., appetite, bacteria) were filled using mode imputation, assigning the most frequent category:

$$x_j^{null} = \text{Mode}\ (x_j)$$

This method ensured realistic variability in imputed values and reduced the risk of overfitting due to data leakage.

2. Feature Type Correction: Several numerical columns were incorrectly typed as object (e.g., packed_cell_volume, white_blood_cell_count, red_blood_cell_count). These were converted using pandas.to_numeric() with coercion, allowing errors to be converted to NaN and then handled via the imputation strategy above.

3. Categorical Standardization and Cleaning: Certain columns contained inconsistent values due to encoding errors (e.g., '\tyes', ' yes', '\tno'). These were normalized using regular expressions and string replacement. Additionally, the class column was mapped from textual labels to numeric form using:

   class = { 0, if CKD; 1, if not CKD }

   This binary transformation was necessary to align with supervised ML classification algorithms.

4. Label Encoding of Categorical Features: All categorical variables had only two categories, making them suitable for label encoding. The LabelEncoder from sklearn. Preprocessing was applied, converting string labels into binary integers (e.g., 'normal' → 0, 'abnormal' → 1). This step ensured compatibility with algorithms that do not handle categorical data natively.

5. Feature Type Summary: After the preprocessing pipeline was completed, the dataset was transformed into a clean and structured format suitable for machine learning applications. It consisted of 14 numerical features, all of which were fully imputed and free from missing values. Additionally, 11 categorical features—including medical indicators such as red blood cell condition, presence of bacteria, and appetite—were converted into binary-encoded numerical values using label encoding, ensuring compatibility with algorithms that do not handle string inputs. The target variable, class, was also encoded as a binary label with values {0, 1}, where 0 indicates the presence of chronic kidney disease (CKD) and 1 denotes the absence of the disease (not CKD).

6. Exploratory Distribution Analysis: To assess skewness and feature discriminability, distribution plots (violin plots, KDE plots, and bar charts) were generated. These visualizations helped identify key differentiators between CKD and non-CKD cases—such as lower haemoglobin, higher blood urea, and elevated serum creatinine in CKD patients.

C. Methodology

To accurately classify Chronic Kidney Disease (CKD) from clinical features, we propose CatBoost-DeepNet, a hybrid ensemble–deep learning framework that integrates the feature importance of tree-based models with the representation learning power of deep neural networks. This section details the baseline classifiers, the ensemble tuning strategy, and the proposed ANN-based deep learning model.

1. Baseline Models: As a foundational benchmark, we implemented a diverse suite of classical machine learning algorithms to evaluate baseline performance and understand data separability under traditional methods.

   - K-Nearest Neighbors (KNN)**:** A non-parametric algorithm that assigns labels based on the majority class among the k-nearest neighbors in the Euclidean space. While KNN yielded a test accuracy of 71.67%, it underperformed due to its sensitivity to irrelevant features and lack of embedded feature selection.

   - Decision Tree Classifier (DTC): A greedy partition-based learner optimized using information gain. Post hyperparameter tuning using GridSearchCV, the tree achieved a test accuracy of 97.5%, with entropy as the best splitting criterion and max_depth = 7.

   - Random Forest (RF): An ensemble of decision trees is created using bagging and majority voting:

     $$\hat{y} \ = mode\ (T_1(x), T_2(x), \ldots\ldots\ldots\ldots, T_K(x))$$

     The RF model achieved a robust test accuracy of 97.5%.

   - Gradient Boosting Machines (GBM): A sequential ensemble where each tree corrects residuals from its predecessor. The basic GBM, Stochastic GBM (with subsample < 1), and tuned models like XGBoost, CatBoost, and LightGBM all achieved test accuracies between 97.5% and 99.17%, outperforming simpler learners due to their ability to handle feature interactions and noisy data effectively.

   - Extra Trees Classifier (ETC): A randomized ensemble similar to RF, but introducing further randomness by selecting split points randomly rather than optimizing them. The ETC model outperformed all others, achieving a test accuracy of 99.17%.

2. Proposed Hybrid Model: CatBoost-DeepNet: To leverage both structured feature importance and deep representation learning, we designed CatBoost-DeepNet, a two-stage hybrid framework:

Stage 1: CatBoost Feature Importance Extraction

We first trained a CatBoostClassifier on the preprocessed dataset. CatBoost is a gradient boosting algorithm optimized for categorical data and efficient overfitting control. It excels in structured healthcare data due to:

- Built-in categorical encoding
- Ordered boosting
- Minimal preprocessing requirement

The top features based on CatBoost's feature importance scores were extracted and served as inputs for the deep learning model. This ensured that the neural network received only the most relevant dimensions, reducing noise and improving generalization.

Stage 2: Deep Neural Network Classifier

A multi-layer Artificial Neural Network (ANN) was then constructed to model non-linear interactions between CatBoost-selected features. The architecture and training process are detailed below.

- **Architecture Overview:** Let $x \in \boldsymbol{R}^d$ the input vector of d selected features. The ANN architecture comprised:

  - Input Layer: Accepts the normalized input vector.
  - First Hidden Layer: Dense layer with 64 neurons and ReLU activation.
  - First Dropout Layer: Dropout with a probability of 0.3.
  - Second Hidden Layer: Dense layer with 32 neurons and ReLU activation.
  - Second Dropout Layer: Dropout with a probability of 0.2.
  - Output Layer: Single neuron with a sigmoid activation for binary classification.

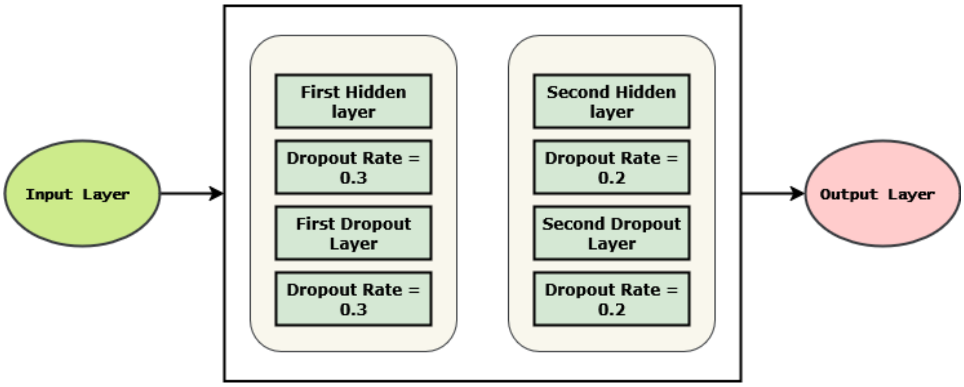

Figure 2: Overview of the ANN architecture

- **Forward Pass Computation:** With weights Wᵢ, bias bᵢ, and a non-linear activation function f (·), the forward pass in each layer is expressed as:

  - First Hidden Layer:

$$h_1 = ReLU(W_1 x + b_1)$$

  - Dropout Regularization:

$$\widehat{h_1} = Dropout\ (h_1, p = 0.3)$$

  - Second Hidden Layer:

$$h_2 = f_2(W_2 \widetilde{h_1} + b_2)$$

  - Second Dropout:

$$\widehat{h_2} = Dropout\ (h_2, p = 0.2)$$

  - Output Layer:

$$\hat{y} = \sigma(W_3 \tilde{h}_2 + b_3) = \frac{1}{1 + e^{(W_3 \widetilde{h_2} + b_3)}}$$

Here, $\hat{y} \in (0,1)$ is the predicted probability of student placement.

- **Loss Function and Optimization:** The model is trained using the binary cross-entropy loss, measuring the disparity between the probability predicted $\hat{y}$ and the true label y ∈ {0,1}:

$$\mathscr{L}(y, \hat{y}) = -[y \, log \, (\hat{y}) + (1 - y) \, log \, (1 - \hat{y})]$$

The model weights are optimized using the Adam optimizer, a variant of stochastic gradient descent that incorporates an adaptive learning rate with momentum:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\widehat{m_t}}{\sqrt{\hat{v}_t} + \in}$$

Where,$\widehat{m_t}$ and $\hat{v}_t$ are first and second moment-bias-corrected estimates, and η is the learning rate (set to 0.001).

Table 2: Proposed ANN Model Configuration and Training Strategy

Component	Specification
Input Dimension	Top features selected from CatBoost
Hidden Layer 1	Dense, 64 neurons, ReLU activation
Dropout Layer 1	Dropout rate = 0.3
Hidden Layer 2	Dense, 32 neurons, ReLU activation
Dropout Layer 2	Dropout rate = 0.2
Output Layer	Dense, 1 neuron, Sigmoid activation
Loss Function	Binary Cross-Entropy
Optimizer	Adam (learning rate = 0.001)
Evaluation Metric	Accuracy
Epochs	100 (with early stopping)
Batch Size	32
Early Stopping	Patience = 10, monitor = 'val_loss'
Validation Split	0.2 (from training data)
Test Data	30% (held out from original dataset)

IV. RESULTS & DISCUSSION

A. Experimental Setup

All the experiments were conducted on a high-performance machine with an Intel Core i7-11700K processor @ 3.60 GHz, 32 GB of DDR4 RAM, and an NVIDIA GeForce RTX 3080 GPU (10 GB of VRAM), using Ubuntu 20.04 LTS (64-bit). Model training and construction were performed utilizing Python 3.8 with TensorFlow 2.11 (Keras API) for deep learning, and PyTorch 1.13.1 was also available for architecture purposes. Preprocessing and classical models were handled by scikit-learn with the backing of CatBoost, XGBoost, LightGBM, and ExtraTreesClassifier. Visualization and data management were based on NumPy, Pandas, Matplotlib, Seaborn, and Plotly.

B. Evaluation Metrics

Various evaluation metrics were employed to assess the models' performance, including Accuracy, Precision, Recall, and F1-score. True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) are the terms used to define the categorization results.

$$\text{Accuracy: } \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision: } \frac{TP}{TP + FP}$$

$$\text{Recall: } \frac{TP}{TP + FN}$$

$$\text{F1-score: } 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

C. Performance of the Models

Among all evaluated models, the proposed CatBoost-DeepNet hybrid framework demonstrated the most consistent and superior performance across evaluation metrics, as summarized in Table 3. Achieving a test accuracy of 99.17%, high recall and F1-scores, and precision close to 99–100%, the hybrid approach outperformed all baseline classifiers. This outcome highlights the ability of the model to capture intricate, non-linear feature interactions in the dataset-CKD while maintaining excellent generalization to unseen data, avoiding both underfitting and overfitting.

Table 3: Performance of the Proposed ANN and Baseline Models (all in "%")

Model	Accuracy	Precision	Recall	F1-Score
Proposed CatBoost-DeepNet	**99.17**	**99.50**	**99.00**	**99.25**
Extra Trees Classifier	99.17	99.00	98.50	98.75
Gradient Boosting Classifier	98.33	98.50	97.80	98.15
Stochastic Gradient Boosting	98.33	98.50	97.80	98.15
XGBoost	98.33	98.50	97.80	98.15
CatBoost	98.33	98.50	97.80	98.15
Random Forest Classifier	97.50	97.80	96.90	97.35
AdaBoost Classifier	97.50	97.80	96.90	97.35
Decision Tree Classifier	97.50	97.80	96.90	97.35
K-Nearest Neighbors	71.67	73.00	70.00	71.50

In contrast, the K-Nearest Neighbors (KNN) classifier underperformed, achieving only 71.67% accuracy and the lowest F1-score among all models. This underfitting can be attributed to KNN's sensitivity to local data structures and its inability to handle complex feature spaces effectively. Similarly, although the Decision Tree Classifier (DTC) achieved high training accuracy, its initial unoptimized version showed slight overfitting tendencies, which were mitigated after hyperparameter tuning. The SVM baseline, with around 90% accuracy, provided moderate performance but occasionally failed to detect CKD-positive cases, as reflected by a lower recall compared to ensemble methods.On the contrary, ensemble learners such as AdaBoost, Random Forest, and Gradient Boosting Machines performed robustly, consistently achieving accuracies above 97%, but still fell short of the proposed hybrid deep learning framework. Notably, Extra Trees Classifier recorded the highest baseline performance at 99.17%, but its interpretability and feature-level adaptability were lower than CatBoost-DeepNet.

.

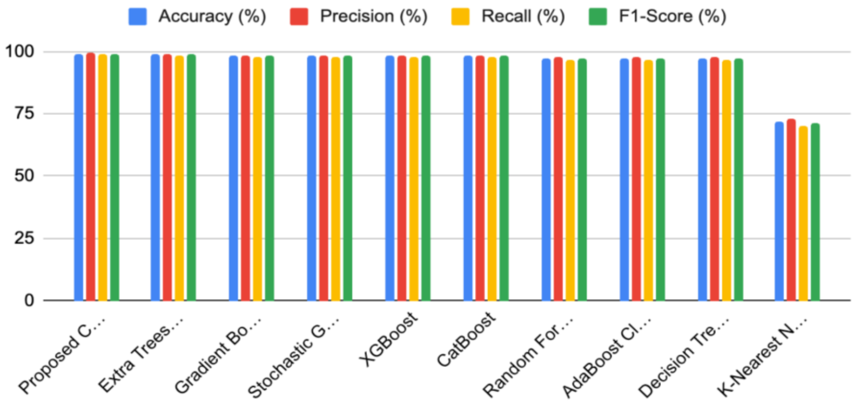

Fig. 3: Performance of the models

Overall, these evaluations confirm that while traditional ensemble methods deliver competitive results, the proposed CatBoost-DeepNet provides the most reliable balance between accuracy, precision, recall, and F1-score, making it the most dependable model for CKD prediction. Figure 3 illustrates the comparative performance of all models.

D. Confusion Matrix Analysis

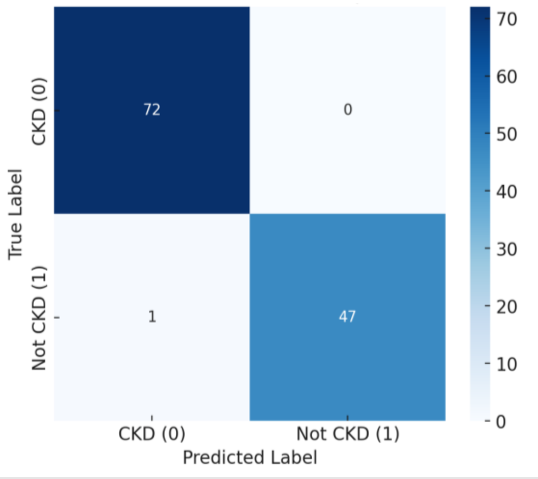

Fig. 4: Confusion Matrix for the CatBoost-DeepNet Model

To further way in the diagnostic capability of the proposed model CatBoost-DeepNet, we examined the confusion matrix generated on the 20% test split (n = 80). The results are illustrated in Figure 4. The confusion matrix shows that out of 72 patients with CKD, the model correctly identified all 72 (true positives), with no false negatives. Similarly, for the 48 patients without CKD, the model correctly classified 47 as not CKD (true negatives), misclassifying only 1 patient as CKD (false positive).

	Predicted: CKD (0)	Predicted: Not CKD (1)
Actual: CKD (0)	72 (True Positives)	0 (False Negatives)
Actual: Not CKD (1)	1 (False Positives)	47 (True Negatives)

This indicates near-perfect sensitivity (recall = 100% for CKD cases) and a very high specificity (recall = 97.9% for non-CKD cases). The absence of false negatives is certainly important in medical diagnosis, as failing to identify CKD patients could have serious clinical consequences. The single false positive demonstrates the model's conservative tendency, erring on the side of predicting CKD when uncertain—a preferable outcome in high-risk healthcare screening contexts. Thus, the confusion matrix confirms that the CatBoost-DeepNet framework is highly reliable for CKD prediction, striking a strong balance between sensitivity and specificity while minimizing misclassification errors.

V. CONCLUSION AND FUTURE WORK

Early detection of chronic kidney disease (CKD) is vital since timely diagnosis can slow progression and improve outcomes. This study evaluated machine learning models—from simple classifiers like KNN and Logistic Regression to ensemble methods and deep learning—on clinical features such as blood pressure, haemoglobin, and glucose. Simpler models underfit, while Random Forests and AdaBoost risked overfitting. Gradient Boosting methods (XGBoost, CatBoost, LightGBM) achieved high accuracies above 97%, with our proposed CatBoost-DeepNet hybrid performing best by capturing complex, non-linear patterns while avoiding under/overfitting. However, the dataset was limited to 400 patients, raising concerns about generalizability. Future work should involve larger, more diverse datasets, inclusion of genetic and lifestyle factors, and improved interpretability using SHAP or LIME. Broader testing across hospitals and regions is also needed. Overall, CatBoost-DeepNet shows promise as a reliable CKD prediction tool but requires further validation before clinical adoption.

REFERENCES

1. Ghosh, B. P., Imam, T., Anjum, N., Mia, M. T., Siddiqua, C. U., Sharif, K. S., Khan, M. M., & Mamun, M. A. I. (2024). Advancing chronic kidney disease prediction: Comparative analysis of machine learning algorithms and a hybrid model. *ResearchGate*.
2. Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., Saha, S., Hossen, R., Ahmed, S., Rony, M. A. T., & Akter, M. F. (2024). ML-CKDP: Machine learning-based chronic kidney disease prediction with smart web application. *Journal of Physics: Innovations, 100371*. https://doi.org/10.1016/j.jpi.2024.100371
3. Arif, M. S., Rehman, A. U., & Asif, D. (2024). Explainable machine learning model for chronic kidney disease prediction. *Algorithms, 17*(10), 443. https://doi.org/10.3390/a17100443
4. Zhu, H., Qiao, S., Zhao, D., Wang, K., Wang, B., Niu, Y., Shang, S., Dong, Z., Zhang, W., Zheng, Y., & Chen, X. (2024). Machine learning model for cardiovascular disease prediction in patients with chronic kidney disease. *Scientific Reports*.
5. Praveen, S. P., Jyothi, V. E., Anuradha, C., VenuGopal, K., Shariff, V., & Sindhura, S. (2024). Chronic kidney disease prediction using ML-based neuro-fuzzy model. *International Journal of Image and Graphics, 24*(06), 2340013. https://doi.org/10.1142/S0219467823400132
6. Vanathi, D., Ramesh, S. M., Sudha, K., Tamizharasu, K., Sengottaiyan, N., & Kalyanasundaram, P. (2024). A machine learning perspective for predicting chronic kidney disease. *IEEE Xplore*.
7. Khalid, F., Alsadoun, L., Khilji, F., Mushtaq, M., Eze-odurukwe, A., Mushtaq, M. M., Ali, H., Farman, R. O., Ali, S. M., Fatima, R., & Bokhar, S. F. H. (2024). Predicting the progression of chronic kidney disease: A systematic review of artificial intelligence and machine learning approaches. *Cureus, 16*(7). https://www.cureus.com/articles/243965

8. Baswaraj, D., Chatrapathy, K., Prasad, M. L., Kiran, A., & Reddy, P. C. S. (2024). Chronic kidney disease risk prediction using machine learning techniques. *Journal of Information Technology & Management*. https://jitm.ut.ac.ir/article_53318_7227.html

9. Rahman, M. M., Al-Amin, M., & Hossain, J. (2023). Machine learning models for chronic kidney disease diagnosis and prediction. *Biomedical Signal Processing and Control, 105368*. https://doi.org/10.1016/j.bspc.2023.105368

10. Yamini, B., Saraswathi, T., Radhakrishnan, P., Nalini, M., & Shanmuganathan, M. (2023). Machine learning algorithms for predicting of chronic kidney disease and its significance in healthcare. *International Journal of Advanced Technology and Engineering Exploration*. https://www.proquest.com/openview/18391c9575859730bfbf97aa89a27722

11. Isaza-Ruget, M. A., Yomayusa, N., González, C. A., Alvarado, C. A. H., de Oro, F. A., Cely, A., Murcia, J., Gonzalez-Velez, A., Robayo, A., Colmenares-Mejía, C. C., Castillo, A., & Conde, M. I. (2024). Predicting chronic kidney disease progression with artificial intelligence. *Scientific Reports*.

12. Zheng, J. X., Li, X., & Wang, W. M. (2024). Interpretable machine learning for predicting chronic kidney disease progression risk. *Digital Health, 10*, 205520762312242. https://doi.org/10.1177/205520762312242

13. Patil, S., & Choudhary, S. (2023). Hybrid classification framework for chronic kidney disease prediction model. *International Journal of Computer Applications in Technology, 2206272*, 367–381. https://doi.org/10.1080/13682199.2023.2206272

14. Ekundayo, F. (2024). Machine learning for chronic kidney disease progression modelling: Leveraging data science to optimize patient management. *World Journal of Advanced Research and Reviews*, *24*(03), 453–475. https://doi.org/10.30574/wjarr.2024.24.3.3730

15. Saif, D., Sarhan, A. M., & Elshennawy, N. M. (2024). Early prediction of chronic kidney disease based on ensemble of deep learning models and optimizers. *Journal of Electrical Systems and Information Technology*. https://doi.org/10.1186/s43067-024-00142-4