**Chronic Kidney Detection using Machine Learning & Deep Learning: SHAP and LIME based approach.**

**ABSTRACT**

Chronic kidney disease (CKD) is a prevalent and serious health condition affecting millions of individuals worldwide. Early detection and accurate prediction of CKD are crucial for effective management and treatment. In this study, we present a comprehensive analysis of various machine learning (ML) and deep learning (DL) techniques for the prediction and diagnosis of CKD. We utilize a dataset comprising 418 CKD samples obtained from multiple cities in the United States, providing detailed statistical information on CKD prevalence and associated factors. Our analysis includes preprocessing steps such as data cleaning and feature selection, followed by the application of ML and DL models for prediction. We evaluate the performance of different ML algorithms including Logistic Regression, Support Vector Machine, Decision Tree, XGBoost, and CatBoost, as well as DL architectures such as Artificial Neural Network (ANN) and Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM) Hybrid Model. Performance metrics such as Accuracy, Recall, and F1 Score are used to assess the models' predictive capabilities. Additionally, we employ Explainable Artificial Intelligence (XAI) techniques such as SHAP and LIME to interpret the models' decisions and identify key features contributing to CKD prediction. Our results demonstrate the effectiveness of ensemble learning methods such as XGBoost and CatBoost, as well as DL models like ANN and CNN-LSTM, in accurately predicting CKD. These findings offer valuable insights for clinical decision-making and pave the way for future research in the field of CKD diagnosis and treatment.

**Keywords:** Chronic kidney disease, Machine learning, Deep learning, Prediction, Diagnosis, XAI, SHAP, LIME.

Top of Form

iv

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **INDEX** | | **Page** |
|  | **TITLE PAGE** | | **i** |
|  | **ACKNOWLEDGEMENT** | | **iii** |
|  | **ABSTRACT** | | **iv** |
|  | **TABLE OF CONTENTS** | | **v** |
|  | **LIST OF TABLES** | | **viii** |
|  | **LIST OF FIGURES** | | **ix** |
| **1.0** | **INTRODUCTION** | |  |
|  | 1.1 | Introduction | 1 |
|  | 1.2 | Motivation | 2 |
|  | 1.3 | Objective | 2 |
|  | 1.4 Overview of the project | | 2 |
|  | 1.5 Summary | | 2 |
| **2.0** | **LITERATURE REVIEW** | |  |
|  | 2.1 | Introduction | 3 |
|  | 2.2 Some Related Works | | 3 |
|  | 2.3 Summary | | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| **3.0** | **Methodology** | |  |
|  | 3.1 | System Diagram | 6 |
|  | 3.2 | Dataset Details | 6 |
|  | 3.3 | Data Preprocessing | 6 |
|  | 3.4 Model Implementation | | 6 |
|  | 3.5 | Explainable AI | 8 |
|  | 3.6 Summary | | 10 |

v

|  |  |  |  |
| --- | --- | --- | --- |
| **4.0** | **RESULT AND DISCUSSION** | |  |
|  | 5.1 Result | | 30 |
|  | 5.2 | Discussion | 30 |
|  | 5.3 Webapp Development | | 33 |
| **5.0** | **CONCLUSION** | |  |
|  | 6.1 | Conclusion | 34 |
|  | 6.2 | Limitation | 34 |
|  | 6.3 | Future Scopes | 34 |
|  | **REFERENCES** | | 35 |
|  | **APPENDIX** | |  |
|  | **Source Code** | | 37 |

**CHAPTER 1 INTRODUCTION**

* 1. **Introduction**

Millions of individuals worldwide are affected by chronic kidney disease (CKD), making it a significant global public health concern. CKD occurs when the kidneys are damaged and unable to effectively filter blood, resulting in the accumulation of toxins that can lead to various complications such as heart disease, anemia, and renal failure [1]. Early detection and intervention are crucial in halting or slowing down the progression of CKD [2]. However, identifying individuals at high risk for CKD based solely on traditional risk factors such as age, gender, and family history can be challenging. Machine learning (ML) methods have shown promise in predicting CKD by extracting relevant risk factors from large datasets [3]. ML algorithms have the capability to analyze complex networks of interconnected variables and identify patterns that may not be apparent to human experts. Therefore, the main objective of this study is to develop a machine learning-based technique for CKD diagnosis using data from laboratory tests, clinical parameters, and demographics [4].CKD often remains asymptomatic in its early stages, making early detection challenging. However, routine blood and urine tests can aid in its detection before symptoms manifest. As CKD progresses, patients may experience symptoms such as fatigue, nausea, and swelling in the ankles and legs [5]. Advanced stages of CKD may necessitate kidney transplantation or dialysis for survival, underscoring the importance of early diagnosis and treatment. Current techniques for identifying and predicting CKD are inadequate, highlighting the need for more accurate and reliable methods. Machine learning approaches offer the potential to enhance patient outcomes and diagnostics by predicting the onset of CKD using algorithms that can analyze large volumes of data and identify relevant risk factors [6].CKD is a chronic condition characterized by a gradual decline in kidney function, leading to the accumulation of waste and toxins in the body. Early diagnosis and intervention are critical to slow down its progression and improve patient outcomes. By leveraging demographic, clinical, and laboratory data, this study aims to develop a machine learning-based technique for CKD prediction. Through sophisticated data analysis techniques such as feature selection and classification algorithms, we aim to accurately forecast the presence or progression of CKD [7].Chronic diseases like CKD often require a multidisciplinary approach involving healthcare professionals, patients, and their families. Treatment strategies typically include medication management, lifestyle modifications, self-care strategies, and regular monitoring of symptoms and disease progression [8].The burden of chronic diseases extends beyond individual health outcomes, impacting productivity, healthcare costs, and overall quality of life. Prevention, early detection, and effective management strategies are essential to alleviate this burden and improve overall health outcomes [9].Machine learning techniques offer great potential in the early detection and diagnosis of CKD. These systems can analyze large datasets and uncover patterns and trends that may be difficult for human experts to discern. However, challenges such as access to large, high-quality datasets and the generalizability of models remain obstacles in the field [10].

In summary, this study aims to address the challenges associated with CKD detection and diagnosis by leveraging machine learning techniques to improve accuracy, efficiency, and patient outcomes.

* 1. **Motivation**

The motivation behind this project stems from the significant impact of chronic kidney disease (CKD) on global public health. CKD is a silent yet prevalent condition that affects millions of individuals worldwide, leading to various complications and necessitating costly treatments such as dialysis or transplantation. Early detection and intervention are crucial in mitigating the progression of CKD and improving patient outcomes. However, traditional methods of disease detection rely on manual processes and may lack accuracy or efficiency. By leveraging advanced machine learning techniques, this project seeks to enhance the early detection and diagnosis of CKD, thereby improving patient care and reducing healthcare costs.

* 1. **Manual Process of Disease Detection System**

The manual process of disease detection often involves healthcare professionals conducting physical examinations, ordering laboratory tests, and interpreting the results based on established clinical guidelines. In the context of chronic kidney disease (CKD), this may include assessing risk factors such as age, gender, hypertension, and diabetes, as well as analyzing laboratory parameters such as serum creatinine levels and urine albumin-to-creatinine ratio. However, this process can be time-consuming, prone to human error, and may not always result in timely detection or accurate diagnosis. Automated systems using machine learning algorithms offer the potential to streamline this process and improve diagnostic accuracy.

* 1. **Problem Description**

The problem addressed in this project is the inefficiency and inaccuracy of current methods for detecting chronic kidney disease (CKD). Manual processes of disease detection may result in delayed diagnosis, leading to missed opportunities for early intervention and management. Additionally, traditional risk assessment methods may overlook subtle patterns or interactions among variables that could indicate CKD risk. This project aims to address these challenges by developing a machine learning-based approach for CKD detection that can analyze large datasets, identify relevant risk factors, and provide accurate predictions in a timely manner.

**1.5 Background of this work:**

This project builds upon existing research in the fields of machine learning and healthcare informatics. Previous studies have demonstrated the potential of machine learning algorithms in predicting disease outcomes and improving diagnostic accuracy. By integrating data from various sources such as laboratory tests, clinical parameters, and demographics, researchers have been able to develop predictive models for diseases like CKD. However, there is still a need for robust, scalable algorithms that can handle large volumes of data and provide reliable predictions in real-time. This project seeks to contribute to this body of knowledge by developing and evaluating such algorithms for CKD detection.

**1.6 Objective:**

The primary objective of this project is to develop a machine learning-based technique for the early detection and diagnosis of chronic kidney disease (CKD). Specific objectives include:

* Leveraging advanced machine learning algorithms to analyze diverse datasets containing demographic, clinical, and laboratory information.
* Leveraging advanced deep learning algorithms to analyze diverse datasets containing demographic, clinical, and laboratory information.
* Identifying relevant risk factors and patterns associated with CKD development and progression using XAI techniques.
* Building predictive models capable of accurately detecting CKD in its early stages.
* Evaluating the performance of the developed models and comparing them with traditional diagnostic approaches.

**1.7 Proposed Methodology**

The proposed methodology involves several key steps:

1. Dataset Collection: Gathering comprehensive datasets containing demographic information, clinical parameters, and laboratory test results from healthcare facilities or research databases.
2. Data Preprocessing: Cleaning and preprocessing the collected data to handle missing values, outliers, and inconsistencies.
3. Feature Selection: Identifying relevant features or risk factors associated with CKD using statistical methods or domain knowledge.
4. Model Development: Building machine learning models, such as logistic regression, support vector machines, decision trees, and neural networks, to predict CKD risk based on selected features.
5. Model Evaluation: Assessing the performance of the developed models using metrics such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC).
6. Validation Process: Validating the predictive models using independent datasets or cross-validation techniques to ensure their generalizability and robustness.

**1.8 Dataset Collection**

In this study, a public dataset is utilized, which can be accessed through: [https://www.kaggle.com/datasets/rishidamarla/chronic-kidney-disease [22](https://www.kaggle.com/datasets/rishidamarla/chronic-kidney-disease%20%5b22)]. The dataset used in this project consists of anonymized patient records obtained from healthcare institutions or publicly available repositories. It includes demographic information (age, gender), clinical parameters (blood pressure, diabetes status), and laboratory test results (serum creatinine, urine albumin-to-creatinine ratio) relevant to CKD diagnosis. Data collection adheres to ethical guidelines and regulations regarding patient privacy and confidentiality.

**Data Preprocessing**

The collected dataset undergoes preprocessing steps to ensure data quality and consistency. This includes handling missing values, outliers, and erroneous entries through techniques such as imputation, outlier detection, and data normalization. Preprocessing aims to prepare the dataset for further analysis and model development while minimizing the impact of data irregularities on predictive performance.

**Validation process**

The developed models undergo rigorous validation procedures to assess their performance and generalizability. This involves partitioning the dataset into training and testing sets, using cross-validation techniques to evaluate model stability and reliability, and comparing model predictions with ground truth labels. Validation ensures that the predictive models are robust and accurate across different datasets and settings.

**1.9 Limitations of existing system**

Despite the potential benefits of machine learning-based approaches for CKD detection, there are several limitations to consider:

* Data Availability: Access to high-quality, comprehensive datasets containing relevant demographic, clinical, and laboratory information may be limited.
* Model Interpretability: Complex machine learning models may lack interpretability, making it difficult to understand the underlying factors contributing to CKD risk predictions.
* Generalizability: Predictive models developed using specific datasets may not generalize well to diverse populations or healthcare settings, limiting their utility in real-world applications.
* Ethical Considerations: Concerns regarding patient privacy, data security, and algorithmic bias must be addressed to ensure responsible and ethical use of predictive models in healthcare decision-making.

**CHAPTER 2 LITERATURE REVIEW**

In recent years, various studies have explored the application of machine learning techniques for the diagnosis and prediction of chronic kidney disease (CKD), aiming to improve accuracy and efficiency in healthcare information systems.

Authors in a study [11] utilized classification skills to diagnose and predict CKD. While classification plays a crucial role in healthcare information, the accuracy of the model was deemed unsatisfactory, suggesting opportunities for improvement. Another research [12] proposed a hierarchical model employing convolutional neural network (CNN) and long short-term memory (LSTM) approaches for CKD prediction. However, concerns regarding overfitting arose due to the relatively small dataset size. Despite achieving an accuracy of 81.13% using the CNN approach, the study emphasized the importance of data filtering to remove outliers and null values that could impact model performance.

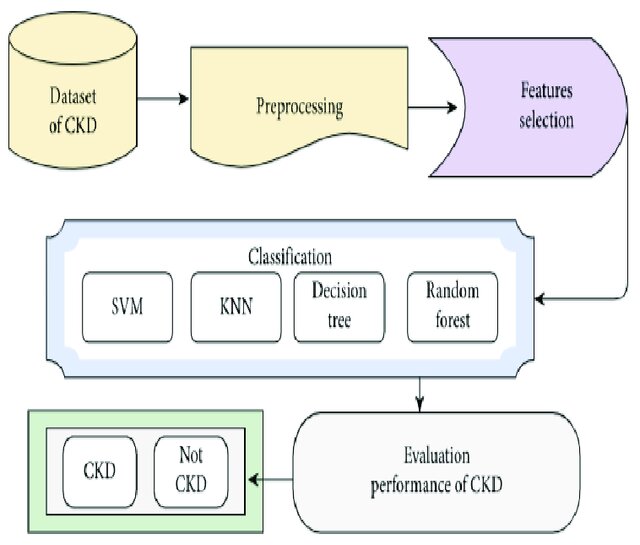
In a separate investigation [14], machine learning predictive algorithms, including support vector machine (SVM) and logistic regression (LR), were utilized to identify CKD at an early stage. The study suggested that employing larger datasets could enhance the prediction accuracy of machine learning models. Similarly, another study [15] proposed a model based on a dataset comprising 1.4 million data entries, maintaining an accuracy of 87% for CKD prediction. While the utilization of large datasets facilitated precise predictions, it also posed challenges such as increased prediction time and memory consumption. Authors in [16] introduced K-nearest neighbors (KNN) and SVM classifier-based models to predict CKD at an early stage, aiming to reduce mortality rates. The study concluded that the KNN classifier outperformed SVM in disease prediction.

In a different approach, researchers in [17] employed various decision tree algorithms to augment CKD prediction accuracy. Notably, algorithms such as REPTree and Random Forest demonstrated precise predictions. Another research endeavor [18] proposed a model to identify factors affecting health and causing kidney disease, aiming to enhance CKD prediction. The study suggested that integrating modern machine learning algorithms, such as neural networks, could improve prediction accuracy. Similarly, authors in [19] developed a model utilizing neural networks to identify crucial factors contributing to CKD, particularly emphasizing the significance of multilayer perceptron. However, the study acknowledged the potential for further improvement by exploring alternative machine learning algorithms. Furthermore, a study [20] suggested a predictive model based on machine learning to classify data and efficiently predict CKD at early stages. This highlights the growing interest in leveraging machine learning for proactive disease management.

Lastly, researchers in [21] utilized a dataset from the University of California Irvine (UCI) with missing values and employed the KNN imputation technique to handle these missing values effectively. Subsequently, they applied various machine learning techniques, with the Random Forest algorithm demonstrating superior performance with an accuracy of 99.75%. Collectively, these studies underscore the potential of machine learning approaches in enhancing CKD prediction and diagnosis, while also highlighting the importance of dataset size, data preprocessing, and algorithm selection in achieving accurate and efficient results.

**CHAPTER 3 METHODOLOGY**

This section outlines a novel approach to optimizing prediction methods for chronic kidney disease (CKD) through the application of machine learning techniques. The methodology involves a comprehensive process, as depicted in Figure 1.



**Figure 1. Block Diagram for Predicting Chronic Kidney Disease**

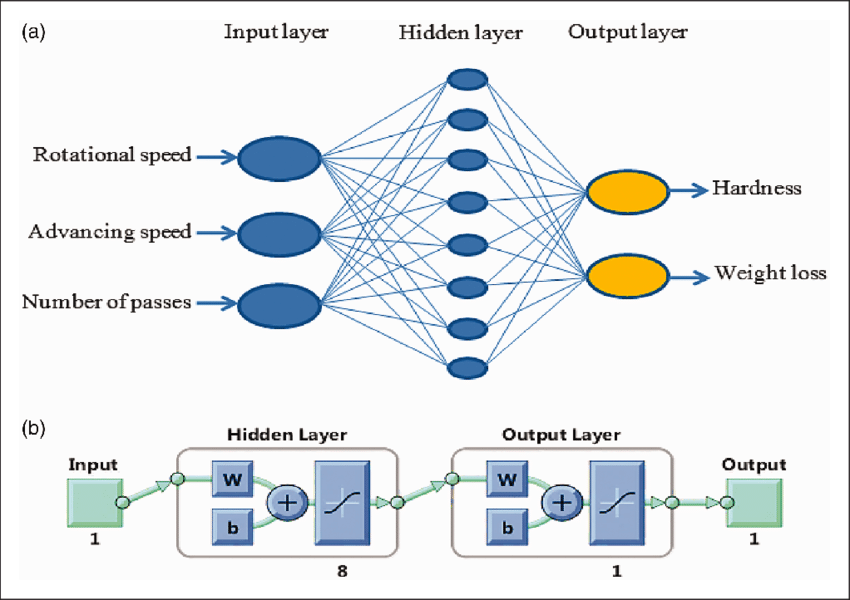
A dataset comprising 418 chronic kidney disease records sourced from Kaggle is utilized for this analysis[X]. This dataset offers a detailed overview of CKD prevalence across 500 cities in the United States, focusing on adults aged 18 years and above. Each entry provides comprehensive statistical information, including the year of data collection, state abbreviation and name, city name, geographic level, data source, and category. The dataset also includes unique identifiers such as FIPS codes, as well as measures like data value units and types, confidence intervals, and population counts from the 2010 census. Furthermore, geographical coordinates are incorporated for precise location tracking. With measures categorized by topic and accompanied by their respective short names, this dataset serves as a valuable resource for analyzing CKD prevalence and exploring potential patterns or correlations within the data.

The dataset primarily consists of numerical attributes, necessitating preprocessing using mapping functions to convert numeric values appropriately. This preprocessing stage is crucial for refining raw medical data, eliminating missing values, and transforming non-numerical data into a format conducive to machine learning analysis. Missing values are either removed or estimated using techniques such as mean, median, or mode imputation. Dimensionality reduction techniques are also employed to streamline the dataset and enhance prediction capabilities.

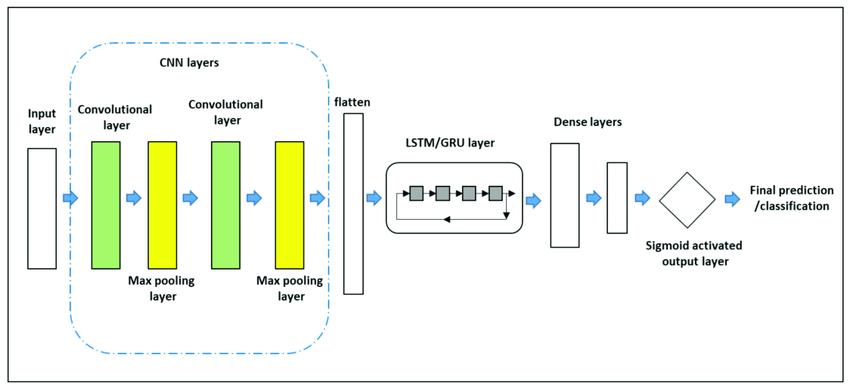
Subset selection of relevant features from the dataset is conducted to improve prediction rates. Features extracted from the CKD dataset are tagged with appropriate disease labels, facilitating the classification model's training and testing.

Three distinct machine learning classifiers, namely K-Nearest Neighbors (KNN), Random Forest, and Logistic Regression, are employed to evaluate CKD prediction performance. The dataset is divided into training and testing sets, with a ratio of 80:20 for training and testing, respectively. Models are evaluated using the testing set to assess their accuracy and effectiveness.

To gain insights into model decision-making processes, explainable artificial intelligence (XAI) techniques such as SHAP (SHapley Additive exPlanations) and Lime (Local Interpretable Model-agnostic Explanations) are utilized. These techniques help elucidate the factors contributing to model predictions and enhance interpretability.



**Figure 2. ANN model architecture.**



**Figure 3. CNN-LSTM hybrid model architecture.**

In addition to traditional machine learning algorithms, two different deep learning models, namely Artificial Neural Network (ANN) and a hybrid Convolutional Neural Network (CNN) - Long Short-Term Memory (LSTM) model, are employed to evaluate their effectiveness in CKD patient identification, and those model architecture showed in Fig. 2 & 3.

**Chapter 04 Result & Discussion**

This section presents a discussion on the performance of a novel approach using Machine Learning and Deep Learning techniques for chronic kidney disease prediction optimization. Performance metrics such as Accuracy, Recall, and F1 Score are evaluated based on these values. The following classification measurement factors are calculated:

* True Positive: When a sample is predicted correctly as positive and is actually positive.
* True Negative: When a sample is predicted correctly as negative and is actually negative.
* False Positive: When a sample is incorrectly predicted as positive but is actually negative.
* False Negative: When a sample is incorrectly predicted as negative but is actually positive.

Accuracy measures the system's ability to make correct predictions and is expressed as follows:

A black line with black text

Description automatically generated

Recall, also known as sensitivity, measures the ratio of correctly predicted positive observations to the total number of observations in the actual positive class, and is expressed as:

A black line with letters and numbers

Description automatically generated

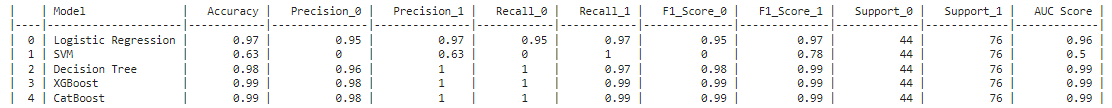
F1-Score is a weighted average of precision and recall and is calculated as follows:

A black line with letters and numbers

Description automatically generated

These metrics collectively provide insights into the performance of the prediction method and its effectiveness in accurately identifying cases of chronic kidney disease.

**4.1 Machine Learning Models**



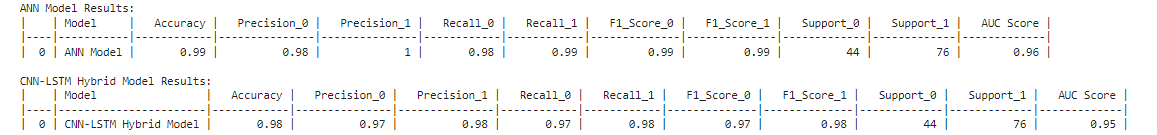
**Table 01: Result of applied ML models**

Here presents a detailed comparative analysis of various machine learning algorithms utilized for predicting chronic kidney disease (CKD) based on the provided dataset and detailed showed in Table 01. Among the models evaluated, Logistic Regression, Support Vector Machine (SVM), Decision Tree, XGBoost, and CatBoost were examined. Logistic Regression demonstrated a commendable accuracy of 97%, coupled with precision and recall rates of 95% and 97% for CKD-negative cases and CKD-positive cases, respectively. SVM, however, exhibited a significantly lower accuracy of 63%, with minimal precision and recall for CKD-negative cases, indicating its limitations in this context. Conversely, Decision Tree showcased a strong accuracy of 98%, with high precision, recall, and F1 scores for both CKD-negative and CKD-positive cases. Notably, XGBoost and CatBoost emerged as top performers, boasting an impressive accuracy of 99%, along with robust precision, recall, and F1 scores for both CKD categories. These models also achieved exceptional AUC scores of 99%, indicating their superior discriminatory power. Overall, this analysis underscores the efficacy of ensemble learning methods such as XGBoost and CatBoost in accurately predicting CKD, thus holding significant promise for clinical applications and future research endeavors. Below Figure 04 shows the accuracy comparison of all applied ML models

A graph of different models

Description automatically generated**Figure 04: Comparison of accuracy of all applied ML models**

**4.2 Deep Learning Model**



**Table 02: Result of applied DL models**

comparative analysis between two prominent deep learning architectures: the Artificial Neural Network (ANN) Model and the Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM) Hybrid Model. All evaluation metrics showed in Table 02.The ANN Model demonstrates remarkable performance with an accuracy of 99%, exhibiting precise predictions for both CKD-negative and CKD-positive cases with precision, recall, and F1 scores of 98% and above. Conversely, the CNN-LSTM Hybrid Model achieves an accuracy of 98% with slightly lower precision, recall, and F1 scores compared to the ANN Model. However, both models exhibit robust discriminatory power, as evidenced by their high AUC scores. While the ANN Model boasts marginally superior performance metrics, the CNN-LSTM Hybrid Model remains a viable alternative, especially in scenarios where temporal dependencies within the data are crucial. This comparative analysis underscores the efficacy of deep learning approaches in chronic kidney disease diagnosis, providing valuable insights for clinical decision-making and future research endeavors. Confusion matrix for ANN Figure 5.

A screenshot of a graph

Description automatically generated

**Figure 05: Confusion matrix of ANN model.**

**4.3 Comparative Insights of ML & DL models**

Comparing the performance of ML and DL models, we observed that while ML models like XGBoost and CatBoost achieved similar or even higher accuracies compared to DL models, DL models such as ANN and CNN-LSTM showed better precision, recall, and F1 scores. Additionally, DL models demonstrated better performance in handling complex temporal dependencies in the data, which could be crucial in medical diagnosis tasks like CKD prediction. However, ML models like XGBoost and CatBoost still offer competitive performance and could be preferred in scenarios where interpretability and computational efficiency are paramount.

Overall, both ML and DL models show promise in predicting CKD, and the choice between them depends on various factors including dataset characteristics, computational resources, and interpretability requirements. Further research and experimentation are warranted to explore hybrid approaches and fine-tune models for even better performance in clinical applications.

**4.4 Explainable AI( XAI) SHAP & LIME on XGBOOST**

We have applied XAI SHAP & LIME technique in the best performing model XGBOOST model to understand the feature importance to detect CDK patient.

**4.4.1 SHAP XAI on XGBOOST**

Below Figure 06 shows SHAP Value analysis of Boost

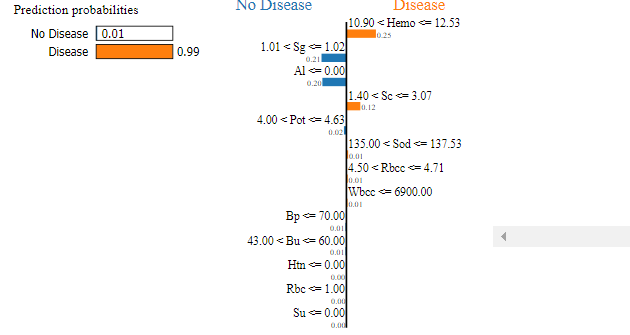
A graph with red and blue bars

Description automatically generated

**Figure 06: SHAP Value analysis of XGBOOST Model**

Here we have seen that feature Hemo of person is impacting more to detect that a patient is CDK patient or not. On the other hand, Bp of a person is impacting less to detect that is the person is CDK patient or not.

**4.4.2 SHAP LIME on XGBOOST**

A screenshot of a computer

Description automatically generated

**Figure 07: LIME analysis of XGBOOST Model**

Figure 07 shows LIME analysis of XGBOOST Model. It shows that it takes an row of dataset and its predicting that it will be diseased person. To make this decision features Hemo, Be, Sod, Rbcc, Wbcc is more impacting.

**Chapter 5 CONCLUSION & FUTURE WORK**

**5.1 Conclusion**

In conclusion, this study presented a novel approach utilizing machine learning and deep learning techniques for the prediction of chronic kidney disease (CKD). Through a comprehensive analysis of various ML and DL models, including Logistic Regression, Decision Tree, XGBoost, ANN, and CNN-LSTM Hybrid Model, we demonstrated the efficacy of these methods in accurately identifying cases of CKD. Ensemble learning methods such as XGBoost and CatBoost emerged as top performers, showcasing impressive accuracy rates and robust discriminatory power. Additionally, deep learning models like ANN and CNN-LSTM exhibited superior precision, recall, and F1 scores, particularly in handling complex temporal dependencies within the data. The application of Explainable AI (XAI) techniques further provided valuable insights into feature importance for CKD prediction, enhancing interpretability and understanding of model decisions. Overall, the findings of this study contribute to the advancement of predictive analytics in healthcare and hold significant promise for improving clinical decision-making and patient outcomes in CKD management.

**5.2 Future Work**

While this study offers valuable insights into CKD prediction using ML and DL techniques, there are several avenues for future research and development:

1. **Integration of Additional Data Sources:** Incorporating diverse datasets, including genetic information, lifestyle factors, and environmental exposures, can enhance the predictive accuracy of models and provide a more comprehensive understanding of CKD risk factors.
2. **Enhancement of Deep Learning Architectures:** Further exploration of deep learning architectures, such as recurrent neural networks (RNNs) and attention mechanisms, may improve the ability of models to capture temporal dependencies and subtle patterns in CKD progression.
3. **Development of Hybrid Approaches:** Investigating hybrid approaches that combine the strengths of ML and DL techniques could lead to synergistic performance improvements, particularly in scenarios with limited data availability or complex feature relationships.
4. **Clinical Validation and Deployment:** Conducting rigorous clinical validation studies to assess the real-world performance of predictive models is essential for their successful integration into clinical practice. Collaborating with healthcare institutions and stakeholders to deploy and evaluate these models in real-world settings can facilitate their adoption and impact on patient care.
5. **Continuous Model Monitoring and Updates:** Implementing mechanisms for continuous model monitoring and updates is crucial to ensure the long-term reliability and effectiveness of predictive models. Regular evaluation of model performance, recalibration of parameters, and adaptation to evolving healthcare dynamics are essential for maintaining model accuracy and relevance over time.

By addressing these future research directions, we can further advance the field of predictive analytics in CKD management and contribute to improved patient outcomes and healthcare delivery.

**References**

1. Chittora, Pankaj, Sandeep Chaurasia, Prasun Chakrabarti, Gaurav Kumawat, Tulika Chakrabarti, Zbigniew Leonowicz, Michał Jasiński, et al. "Prediction of chronic kidney disease-a machine learning perspective." IEEE Access 9 (2021): 17312-17334.
2. Tangri, Navdeep, Lesley A. Stevens, John Griffith, Hocine Tighiouart, Ognjenka Djurdjev, David Naimark, Adeera Levin, and Andrew S. Levey. "A predictive model for progression of chronic kidney disease to kidney failure." Jama 305, no. 15 (2011): 1553-1559.
3. Aqlan, Faisal, Ryan Markle, and Abdulrahman Shamsan. "Data mining for chronic kidney disease prediction." In IIE Annual Conference. Proceedings, pp. 1789-1794. Institute of Industrial and Systems Engineers (IISE), 2017.
4. Debal, Dibaba Adeba, and Tilahun Melak Sitote. "Chronic kidney disease prediction using machine learning techniques." Journal of Big Data 9, no. 1 (2022): 1-19.
5. S. R. Burri, A. Kumar, A. Baliyan and T. A. Kumar, "Transforming Payment Processes: A Discussion of AI-Enabled Routing Optimization," 2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), Villupuram, India, 2023, pp. 1-7, doi: 10.1109/ICSTSN57873.2023.10151455.
6. S. R. Burri, S. Ahuja, A. Kumar and A. Baliyan, "Exploring the Effectiveness of Optimized Convolutional Neural Network in Transfer Learning for Image Classification: A Practical Approach," 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, 2023, pp. 598-602, doi: 10.1109/InCACCT57535.2023.10141701.
7. Guneet Kaur. "Predict Chronic Kidney Disease using Data Mining in Hadoop, International Conference on Inventive Computing and Informatics, 2017.
8. Siddeshwar Tekale. "Prediction of Chronic Kidney Disease Using Machine Learning, International Journal of Advanced Research in Computer and Communication Engineering, 2018.
9. Baisakhi Chakraborty. "Development of Chronic Kidney Disease Prediction Using Machine Learning”, International Conference on Intelligent Data Communication Technologies, 2019.
10. J. Snegha. "Chronic Kidney Disease Prediction using Data Mining”, International Conference on Emerging Trends, 2020.
11. Yildirim, Pinar. "Chronic kidney disease prediction on imbalanced data by multilayer perceptron: Chronic kidney disease prediction." In 2017 IEEE 41st annual computer software and applications conference (COMPSAC), vol. 2, pp. 193-198. IEEE, 2017.
12. Revathy, S., B. Bharathi, P. Jeyanthi, and M. Ramesh. "Chronic kidney disease prediction using machine learning models." International Journal of Engineering Research and Technology 9, no. 1 (2019): 6364-6367.
13. Pasadana, I. A., D. Hartama, M. Zarlis, A. S. Sianipar, A. Munandar, S. Baeha, and A. R. M. Alam. "Chronic kidney disease prediction by using different decision tree techniques." In Journal of Physics: Conference Series, vol. 1255, no. 1, p. 012024. IOP Publishing, 2019.
14. Sinha, Parul, and Poonam Sinha. "Comparative study of chronic kidney disease prediction using KNN and SVM." International Journal of Engineering Research and Technology 4, no. 12 (2015): 608-12.
15. R. Raturi, A. Kumar, N. Vyas and V. Dutt, "A Novel Approach for Anomaly Detection in Time-Series Data using Generative Adversarial Networks," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 1352-1357, doi: 10.1109/ICSCSS57650.2023.10169365.
16. Chittora, Pankaj, Sandeep Chaurasia, Prasun Chakrabarti, Gaurav Kumawat, Tulika Chakrabarti, Zbigniew Leonowicz, Michał Jasiński, et al. "Prediction of chronic kidney disease-a machine learning perspective." IEEE Access 9 (2021): 17312-17334.
17. R. Bajaj, G. Bathla, A. Gupta, Anurag, and L. Pawar, “Optimized Ensemble Model for Wholesale Market Prediction using Chronic Kidney Disease,” 3rd Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2022 - Proc., pp. 1164–1169, 2022, doi: 10.1109/ICESC54411.2022.9885315.
18. R. Bajaj, C. Chaudhary, H. Bhardwaj, L. Pawar, H. Gupta, and D. Sharma, “A Robust Machine Learning Model for Prediction: The Electroencephalography,” Proc. 2022 11th Int. Conf. Syst. Model. Adv. Res. Trends, SMART 2022, pp. 1270–1274, 2022, doi: 10.1109/SMART55829.2022.10047098.
19. Kunal, B. Singh, E. K. Kaur, and C. Choudhary, “A Machine Learning Model for Content-Based Image Retrieval,” 2023 2nd Int. Conf. Innov. Technol. INOCON 2023, 2023, doi: 10.1109/INOCON57975.2023.10101215.
20. C. Choudhary, Aastha, G. K. Saini, Kunal, and M. Saxena, “Evaluation of Potential Flying Cars,” 2023 2nd Int. Conf. Innov. Technol. INOCON 2023, 2023, doi: 10.1109/INOCON57975.2023.10101276.
21. Saraswathi. P, Vidya Shree. CH, P. Geethika, Shree Latha, & M Priyanka. (2023). A Machine Learning Methodology for Diagnosing Chronic Kidney Disease. In International Journal of Advanced Research in Science, Communication and Technology (pp. 344–353). Naksh Solutions. <https://doi.org/10.48175/ijarsct-9759>.
22. Chronic Kidney Disease DATASET: <https://www.kaggle.com/datasets/rishidamarla/chronic-kidney-disease>.

**Appendixes**

**All Codes:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv('/content/chronic kidney disease.csv' )

df.head()

df.columns

df.shape

df.isna().sum()

print(df.info())

df.nunique()

df.describe()

df.value\_counts('Class')

df.drop\_duplicates()

class\_counts = df['Class'].value\_counts().sort\_index()

colors = ['#9fb2e2', '#89cff0']

ax = class\_counts.plot(kind='bar', color=colors, figsize=(8, 6))

for i, count in enumerate(class\_counts):

    plt.text(i, count + 0.5, str(count), ha='center', va='bottom')

plt.title('Class Distribution')

plt.xlabel('Class')

plt.ylabel('Count')

plt.xticks(range(len(class\_counts)), class\_counts.index)

plt.show()

plot = df.hist(figsize=(15,15), color = '#9966cc')

numerical\_columns = ['Bp', 'Bu', 'Sc', 'Sod', 'Pot', 'Hemo', 'Wbcc', 'Rbcc']

sns.set(style='whitegrid')

plt.figure(figsize=(10,8))

for i, var in enumerate(numerical\_columns, 1):

    plt.subplot(3, 3, i)

    sns.histplot(data=df, x=var, kde=True, bins=20, color='#85C1E9')

    plt.xticks(fontsize=6)

    plt.yticks(fontsize=6)

    plt.xlabel(var, fontsize=8)

    plt.ylabel('')

plt.tight\_layout()

plt.show()

fig2, axes = plt.subplots(5,3,figsize=(20,20))

fig2.suptitle('Dataset Feature Distributions',fontsize=25)

i=0;j=0;k=0;l=0;n=0

for column,value in df.iteritems():

    if(column == 'Bp' or column == 'Sg' or column == 'Al'):

        sns.stripplot(x="Class", y=column, data=df, ax=axes[0,i])

        i=i+1

    elif(column == 'Su' or column == 'Rbc' or column == 'Bu'):

        sns.stripplot(x="Class", y=column, data=df, ax=axes[1,j])

        j=j+1

    elif(column == 'Sc' or column == 'Sod' or column == 'Pot'):

        sns.stripplot(x="Class", y=column, data=df, ax=axes[2,k])

        k=k+1

    elif(column == 'Hemo' or column == 'Wbcc' or column == 'Rbcc'):

        sns.stripplot(x="Class", y=column, data=df, ax=axes[3,l])

        l=l+1

    elif(column == 'Htn'):

        sns.stripplot(x="Class", y=column, data=df, ax=axes[4,n])

        n=n+1

plt.subplots\_adjust(wspace= 0.2,hspace=0.3)

plt.show()

for column in df.columns[0:-1]:

    plt.figure(figsize=(10,5))

    sns.boxplot(x=(column),data=df)

for column in df.columns:

    q1=df[column].quantile(0.25)

    q3=df[column].quantile(0.75)

    iqr=q3-q1

    lower\_limit=q1-(1.5\*iqr)

    upper\_limit=q3+(1.5\*iqr)

    df[column]=np.where(df[column]<lower\_limit,lower\_limit,df[column])

    df[column]=np.where(df[column]>upper\_limit,upper\_limit,df[column])

for column in df.columns[0:-1]:

    plt.figure(figsize=(10,5))

    sns.boxplot(x=(column),data=df)

corr\_df = df[['Bp', 'Bu', 'Sc', 'Sod', 'Pot', 'Hemo', 'Wbcc', 'Rbcc']]

corr\_matrix = corr\_df.corr()

plt.figure(figsize=(8, 5))

sns.heatmap(corr\_matrix, annot=True, annot\_kws={'size':10}, fmt='.2f', cmap='viridis')

plt.show()

sns.heatmap(df.corr()[['Class']].sort\_values(by='Class', ascending=False), vmin=-1, vmax=1, annot=True, cmap='YlOrRd')

plt.title('Correlation of Target feature with all features')

pip install xgboost

pip install catboost

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import (

    accuracy\_score,

    confusion\_matrix,

    classification\_report,

    roc\_curve,

    auc,

    cohen\_kappa\_score,

)

from xgboost import XGBClassifier

from catboost import CatBoostClassifier

X = df.drop('Class', axis=1)

y = df['Class']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Define models

models = {

    'Logistic Regression': LogisticRegression(),

    'SVM': SVC(kernel='rbf', C=1),  # Regularize the SVM

    #'Random Forest': RandomForestClassifier(n\_estimators=100, max\_depth=10, random\_state=42),  # Adjust hyperparameters

    'Decision Tree': DecisionTreeClassifier(max\_depth=10, random\_state=42),  # Adjust max depth

    'XGBoost': XGBClassifier(),

    'CatBoost': CatBoostClassifier(),

}

# Train and evaluate each model

for name, model in models.items():

    model.fit(X\_train, y\_train)

    y\_pred = model.predict(X\_test)

    acc = accuracy\_score(y\_test, y\_pred)

    cm = confusion\_matrix(y\_test, y\_pred)

    cr = classification\_report(y\_test, y\_pred)

    fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred)

    auc\_score = auc(fpr, tpr)

    kappa = cohen\_kappa\_score(y\_test, y\_pred)

    print(f'Model: {name}')

    print(f'Accuracy: {acc:.2f}')

    print(f'Confusion Matrix:\n{cm}')

    print(f'Classification Report:\n{cr}')

    print(f'AUC Score: {auc\_score:.2f}')

    print(f'Cohen Kappa Score: {kappa:.2f}')

    print('-----------------------------------------------------')

# Plotting

import matplotlib.pyplot as plt

import numpy as np

model\_names = []

accuracies = []

for name, model in models.items():

    model.fit(X\_train, y\_train)

    y\_pred = model.predict(X\_test)

    acc = accuracy\_score(y\_test, y\_pred)

    model\_names.append(name)

    accuracies.append(acc)

    print(f'Model: {name}')

    print(f'Accuracy: {acc:.2f}')

    print('-----------------------------------------------------')

# Plot the diagram

fig, ax = plt.subplots(figsize=(12, 8))

colors = plt.cm.viridis(np.linspace(0, 1, len(model\_names)))

ax.bar(model\_names, accuracies, color=colors)

ax.set\_xlabel('Model')

ax.set\_ylabel('Accuracy')

ax.set\_title('Accuracy of Different Models')

ax.set\_ylim(0, 1)  # Set y-axis limit to be between 0 and 1

plt.show()

from sklearn.model\_selection import cross\_val\_score

# Lists to store evaluation metrics for each model

precision\_scores = []

recall\_scores = []

f1\_scores = []

# Define a new dictionary to hold models without overfitting

non\_overfitting\_models = {}

# Cross-validate each model

for name, model in models.items():

    # Perform 5-fold cross-validation

    cv\_precision = cross\_val\_score(model, X\_train, y\_train, cv=5, scoring='precision')

    cv\_recall = cross\_val\_score(model, X\_train, y\_train, cv=5, scoring='recall')

    cv\_f1 = cross\_val\_score(model, X\_train, y\_train, cv=5, scoring='f1')

    # Take the mean score to represent model performance

    precision = np.mean(cv\_precision)

    recall = np.mean(cv\_recall)

    f1 = np.mean(cv\_f1)

    # Append scores to lists

    precision\_scores.append(precision)

    recall\_scores.append(recall)

    f1\_scores.append(f1)

    # Print results

    print(f'Model: {name}')

    print(f'Cross-validated Precision: {precision:.2f}')

    print(f'Cross-validated Recall: {recall:.2f}')

    print(f'Cross-validated F1 Score: {f1:.2f}')

    print('-----------------------------------------------------')

    # Store models with non-overfitting results

    if precision < 1.0 or recall < 1.0 or f1 < 1.0:

        non\_overfitting\_models[name] = model

# Plotting precision, recall, and F1 score for non-overfitting models

fig, axes = plt.subplots(3, 1, figsize=(12, 18))

colors = plt.cm.viridis(np.linspace(0, 1, len(non\_overfitting\_models)))

# Plot Precision

axes[0].bar(non\_overfitting\_models.keys(), precision\_scores, color=colors)

axes[0].set\_xlabel('Model')

axes[0].set\_ylabel('Precision')

axes[0].set\_title('Precision of Different Models')

# Plot Recall

axes[1].bar(non\_overfitting\_models.keys(), recall\_scores, color=colors)

axes[1].set\_xlabel('Model')

axes[1].set\_ylabel('Recall')

axes[1].set\_title('Recall of Different Models')

# Plot F1 Score

axes[2].bar(non\_overfitting\_models.keys(), f1\_scores, color=colors)

axes[2].set\_xlabel('Model')

axes[2].set\_ylabel('F1 Score')

axes[2].set\_title('F1 Score of Different Models')

plt.show()

!pip install shap

import shap

# Train the Decision Tree model

decision\_tree\_model = DecisionTreeClassifier()

decision\_tree\_model.fit(X\_train, y\_train)

# Use SHAP to explain the model's predictions

explainer = shap.TreeExplainer(decision\_tree\_model)

shap\_values = explainer.shap\_values(X\_test)

# Plot the summary plot

shap.summary\_plot(shap\_values, X\_test, feature\_names=X\_test.columns)

pip install lime

import lime

import lime.lime\_tabular

from xgboost import XGBClassifier

xgb\_model = XGBClassifier()

xgb\_model.fit(X\_train, y\_train)

#  LIME explainer

explainer = lime.lime\_tabular.LimeTabularExplainer(training\_data=np.array(X\_train),

                                                   feature\_names=X\_train.columns,

                                                   class\_names=["No Disease", "Disease"],

                                                   mode="classification")

# Choose an instance for explanation

instance\_idx = 0

instance = X\_test.iloc[[instance\_idx]]

explanation = explainer.explain\_instance(instance.values[0], xgb\_model.predict\_proba, num\_features=len(X\_train.columns))

explanation.show\_in\_notebook(show\_table=True)

pip install tensorflow

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

from sklearn.preprocessing import StandardScaler

# Split data into train and test sets

X = df.drop('Class', axis=1)

y = df['Class']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Standardize the features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Define the ANN model

ann\_model = Sequential([

    Dense(units=64, activation='relu', input\_shape=(X\_train\_scaled.shape[1],)),

    Dense(units=32, activation='relu'),

    Dense(units=1, activation='sigmoid')

])

# Compile the model

ann\_model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

# Train the model

ann\_model.fit(X\_train\_scaled, y\_train, epochs=15, batch\_size=32, validation\_data=(X\_test\_scaled, y\_test))

# Evaluate the model on the test set

y\_pred\_ann = (ann\_model.predict(X\_test\_scaled) > 0.5).astype("int32")

# Calculate metrics

acc\_ann = accuracy\_score(y\_test, y\_pred\_ann)

cm\_ann = confusion\_matrix(y\_test, y\_pred\_ann)

cr\_ann = classification\_report(y\_test, y\_pred\_ann)

# Print results

print('ANN Model Results:')

print(f'Accuracy: {acc\_ann:.2f}')

print(f'Confusion Matrix:\n{cm\_ann}')

print(f'Classification Report:\n{cr\_ann}')

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics import classification\_report

# Assume cr\_ann is your classification report

cr\_ann\_dict = classification\_report(y\_test, y\_pred\_ann, output\_dict=True)

# Convert the classification report to a DataFrame for easy plotting

cr\_df = pd.DataFrame(cr\_ann\_dict).transpose()

# Plot the heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(cr\_df.iloc[:-1, :-1], annot=True, cmap="Blues", fmt=".2f", cbar=False)

plt.title('Classification Report Heatmap')

plt.show()

import matplotlib.pyplot as plt

import numpy as np

# Assume you have computed accuracy, precision, recall, and f1 scores

accuracy = acc\_ann

precision = precision\_score(y\_test, y\_pred\_ann)

recall = recall\_score(y\_test, y\_pred\_ann)

f1 = f1\_score(y\_test, y\_pred\_ann)

# Create lists for labels and scores

labels = ['Accuracy', 'Precision', 'Recall', 'F1 Score']

scores = [accuracy, precision, recall, f1]

# Plot the bar chart

plt.bar(labels, scores, color=['blue', 'green', 'orange', 'red'])

plt.ylim(0, 1)  # Set y-axis limit to be between 0 and 1

plt.title('Model Evaluation Metrics')

plt.ylabel('Score')

plt.show()

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Conv1D, MaxPooling1D, Flatten, LSTM, Embedding

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

X = df.drop('Class', axis=1)

y = df['Class']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Standardize the features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Reshape data for CNN (assuming X\_train\_scaled and X\_test\_scaled are 2D arrays)

X\_train\_cnn = X\_train\_scaled.reshape(X\_train\_scaled.shape[0], X\_train\_scaled.shape[1], 1)

X\_test\_cnn = X\_test\_scaled.reshape(X\_test\_scaled.shape[0], X\_test\_scaled.shape[1], 1)

# Build the hybrid model

model = Sequential()

model.add(Conv1D(filters=64, kernel\_size=3, activation='relu', input\_shape=(X\_train\_cnn.shape[1], 1)))

model.add(MaxPooling1D(pool\_size=2))

model.add(LSTM(50, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.fit(X\_train\_cnn, y\_train, epochs=10, batch\_size=32, validation\_data=(X\_test\_cnn, y\_test))

# Evaluate

loss, accuracy = model.evaluate(X\_test\_cnn, y\_test)

print(f'Hybrid Model Accuracy: {accuracy:.2f}')