

A COMPREHENSIVE MOBILE APPLICATION TO MONITOR AND ENHANCE KIDNEY HEALTH FOR DIALYSIS PATIENTS USING MACHINE LEARNING

Rachel Sun¹, Yujia Zhang²

¹Brookfield Academy, 3215 N Brookfield Rd, Brookfield, WI 53045

²Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

In the evolving landscape of healthcare, timely and accurate medical predictions are paramount, especially in managing chronic conditions like kidney disease. This paper introduces an innovative AI-driven application designed to enhance renal health management by predicting the need for dialysis and anemia, critical aspects of kidney care. Utilizing advanced algorithms such as Support Vector Machine (SVM) and XGBoost, coupled with cross-validation techniques, the application aims to provide reliable health predictions based on patient data. Challenges including model accuracy and processing speed were meticulously addressed through algorithm optimization and efficient data handling, ensuring the system's responsiveness to varying data complexities. Experimentation with mock patient scenarios revealed the system's capability to deliver precise anemia predictions and identify dialysis needs promptly, highlighting its potential in clinical settings. The application's blend of accuracy, speed, and user-centric design positions it as a valuable tool for patients and healthcare providers, promising to improve outcomes and decision-making in kidney health management.

KEYWORDS

Dialysis, Renal Health, AI, Monitoring

1. INTRODUCTION

Kidney failure, the last and most severe stage of Chronic Kidney Disease [9], is when the kidneys can no longer excrete toxins and filter adequately for the human body [10], posing life threatening risks and leading to the urgent medical intervention of Dialysis [11][12]. Approximately 850 million patients are facing various kidney ailments and issues while one tenth of the global adult population are experiencing Chronic Kidney Disease [5]. In 20th century, infectious diseases influenced global death, but currently, noncommunicable diseases impact people more prevalently, indicated by the majority of kidney failure cases caused by diabetes [8]. With a substantial occurrence rate, kidney failure progresses to be more wide-spreading and frequent rising with age, with 949 annual cases per million for ages of seniors above 80 and below 89 [1][2]. Their conditions are often unstable and sometimes critical. However, the resources needed to conduct diagnosis and dialysis on a frequent basis is costly and places a high financial burden on many patients undergoing kidney failure [4]. In Burkina Faso, patients pay a average monthly cost of 75842 CFA, exceeding the annual minimum

wage of the country of 34664 CFA [3]. Similarly, out of the 800,000 people in Bangladesh who suffer from kidney failure and need dialysis for treatment, only 30,000 can afford and access it [7]. Besides the financial struggles of efficiently and frequently receiving dialysis and forms of medical monitoring, patients also face problems of resource inaccessibility and medical scarcity. Bangladesh only currently provides 101 centers for kidney failure and dialysis treatment and are all primary situated in urban areas [7]. The expense of daily dialysis and the relative inaccessibility of current methods to track kidney health can potentially worsen present conditions for dialysis patients. In some poorer countries, many patients even die at an early stage due to lack of treatment [6].

There are several methods that purposes to enhance patient kidney health during dialysis and Chronic Kidney Disease (CKD). The Global Kidney Health Atlas (GKHA) offers a broad strategy to enhance patient kidney health and treatment efficiency, using data collection methods like surveys [13]. However, it lacks personalized patient care. Our application provides individualized analysis, addressing this limitation and ensuring effective kidney care monitoring. mHealth is a application developed aiming to provide management for chronic kidney disease (CKD), enabling early diagnosis and self-monitoring [14]. Users can track medication, monitor blood pressure and glucose levels, and assess CKD risk factors. However, specific biomarkers like creatinine creates confusion for users and affects accuracy. Our application simplifies prediction scales, focusing only on four key and basic indicators, improving user understanding. The Heart Rate Smart Wristband utilizes wearable technology, a health management platform, and social media to enhance kidney patient care among those with Chronic Kidney Disease (CKD) [15]. However, the study's limited participant pool and device availability are limitations. Our mobile app improves from this issue from ensuring universal access.

We propose a comprehensive mobile application developed to monitor kidney health complementing a doctor's diagnosis and advice to improve the efficiency of medical treatment pertaining to kidney diseases. Integrating an AI algorithm trained through machine learning, the app is dedicated in tracking patient kidney condition through advanced statistical data analysis. One of the app's primary objectives is to detect the presence of anemia within the patient's body, a common and essential concern for dialysis patients. The AI algorithm primarily acts to analyze relevant parameters in the Anemia Prediction Function; including hemoglobin, packed cell volume, red blood cell count, and white blood cell count (the four most prominent indicators for dialysis patients); and identify early signs of anemia. Once identified, the application can promptly alert the patient and their healthcare provider with an instant and data-driven prediction. This enables patients to take immediate preventive measures or adjustments to the treatment plan, such as dialysis initiation or modification. The app can also provide individuals with assistance in tracking the progression and changes in their data values throughout dialysis or kidney failure treatments through the Renal Care Track.

This functions as a calendar for users to enter vital information such as blood pressure, phosphorous, potassium, and many other applicable indicators and empowers patients to assess and manage their own personal health. The app stores inputs to preserve a database foundation for users to view their information and analysis after they login to their accounts. This prevents patients from neglecting hidden kidney failure symptoms resulting in late-stage diagnosis. Patients can also have a reduced financial burden and can utilize limited treatment resources more efficiently.

Experiment A aimed to identify the most accurate AI model for predicting the need for dialysis, focusing on minimizing prediction errors. By evaluating ten models with a dataset from Kaggle and employing cross-validation, we sought to ensure reliability and avoid overfitting. The XGBoost model emerged as the most accurate, likely due to its ability to handle diverse and complex datasets effectively through ensemble learning, which combines multiple weak learners to improve predictive performance.

Experiment B investigated the AI program's speed in processing anemia predictions, crucial for timely healthcare decisions in renal dialysis. Using mock patient data to simulate real-world scenarios, we measured the time from data input to prediction output. The findings revealed a correlation between data complexity and processing time, with more complex scenarios leading to longer processing times. This highlighted the importance of optimizing the model and data processing to achieve quicker response times without compromising accuracy, essential for effective patient care in urgent medical situations.

Both experiments underscored the critical balance between accuracy and speed in developing AI-driven healthcare solutions, emphasizing the need for advanced algorithms like XGBoost and the need for optimization to cater to clinical demands efficiently.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Ensuring the accuracy and reliability

Machine Learning is a vital component in the development of the application as it was utilized to examine the data and analyze predictions. However, ensuring the accuracy and reliability of the anemia prediction feature posed a set of challenges. Acquiring a consistent and accurate estimate in prediction showed some discrepancies from time-to-time as the percentage estimated for the presence of anemia did not provide 100% of accuracy. These variations in accuracy requires thorough testing of different models (e.g. logistic regression, Decision Tree Classifier) and choosing the one that indicates the best predictive capability: the Support Vector Machine.

2.2. Setting up the development tools and integrating software components

Another component in the development process was the utilization of software tools and frameworks. The process of setting up the development tools and integrating software components created various challenges and setbacks. For instance, when working with Android Studio, a significant software tool for Android app development, there were compatibility issues for the device, which made it unusable. This situation prompted the exploration of alternatives, a substituting platform that can also function for app development, such as Flutlab. While Flutlab will not be as equally efficient as Android Studio, considering its slower build time and occasional lags, it offers accessibility and continued progress for the development of the application.

2.3. Creating and integrating a connection between components

Creating and integrating a connection between components was a significant challenge in the development process of the app. The main setback was the variance and difference in language between each component: Flutter for the application and Python for the machine learning section and the server. Therefore, directly connecting the different components was a challenge considering their distinct and contrasting languages. To address this issue, we can use specific libraries to connect the components, most importantly using Flask to set up the server functioning as a link between different components. (Flutter uses HTTP to connect to the server, which functions to load and predict the data and information).

3. SOLUTION

The main structure of the kidney health monitoring application is designed to enhance patient engagement and management of kidney health through technology. The program is built around three major components: User Interface (UI), Machine Learning (ML) Algorithm, and Notification System. User Interface (UI): The application starts with a Login Screen, where users authenticate themselves. Once logged in, they reach the Dashboard, which serves as the central hub for accessing various features, including Anemia Prediction and Renal Care Track. The UI is designed for ease of use, allowing patients to input health data and view their health metrics.

Machine Learning (ML) Algorithm: At the heart of the application is the ML algorithm, particularly utilized in the Anemia Prediction feature. It analyzes input parameters (e.g., hemoglobin levels, red and white blood cell counts) to predict the risk of anemia. The algorithm is trained on clinical data, ensuring its predictions are based on established health patterns and indicators.

Notification System: This component alerts users to significant health updates, including predictions of anemia risk from the ML algorithm. It ensures timely communication of critical health information, facilitating prompt action.

The program's flow is linear and user-centric, moving from authentication (Login Screen) to health monitoring and prediction (Dashboard, Anemia Prediction, Renal Care Track), and culminating in proactive health management through alerts (Notification System).

The application is developed using Flutter for the UI, ensuring cross-platform compatibility, Python for the ML algorithm due to its rich ecosystem of data science libraries, and Flask for setting up a lightweight server that connects the Flutter app with Python-based ML predictions. This blend of technologies ensures a seamless, efficient, and responsive application experience from start to finish. As presented in the Machine Learning section, data is located into only true through the classification of ckd as variables are utilized to separate input and output values that are divided into columns, assigning numbers for each variable. X train provides the data features and Y train determines the presence of anemia through learning points in the dataset. In the graph, X train is plotted together with y train, drawing a line to classify the presence of anemia within the dataset. Similarly, the X test plotted recognizes which section of the line it lies, which offers the Y test a opportunity to check current results for the proportion of accuracy through the Support Vector Machine.



Figure 1. Screenshot of add record

```

# Assuming the SVM model and scaler have been pre-trained and saved as
'anemia_svm_model.joblib' and 'scaler.joblib'
model = load('anemia_svm_model.joblib')
scaler = load('scaler.joblib')

def predict_anemia(hemoglobin, packed_cell_volume, red_blood_cell_count,
white_blood_cell_count):
    """
    Predicts the risk of anemia based on blood parameters.

    Parameters:
    - hemoglobin (float): Hemoglobin level.
    - packed_cell_volume (float): Packed Cell Volume percentage.
    - red_blood_cell_count (float): Red Blood Cell count.
    - white_blood_cell_count (float): White Blood Cell count.

    Returns:
    - prediction (str): 'High risk' or 'Low risk' of anemia.
    """
    # Prepare the input data in the correct format
    input_features = np.array([[hemoglobin, packed_cell_volume, red_blood_cell_count,
white_blood_cell_count]])
    input_features_scaled = scaler.transform(input_features)

    # Make the prediction
    prediction = model.predict(input_features_scaled)

    # Translate the prediction into a more user-friendly output
    if prediction == 1:
        return 'High risk of anemia'
    else:
        return 'Low risk of anemia'

```

Figure 2. Screenshot of code 1

This code snippet demonstrates how to integrate a machine learning model into the backend of a health monitoring application. The `predict_anemia` function encapsulates the process of taking user-inputted blood parameters, scaling them appropriately (as the model was likely trained on scaled data), making a prediction with the SVM model, and then translating that prediction into a human-readable risk assessment.

The server uses flask in python library which sets up the website. With given information of red cell, white cell, hemoglobin, and pact cell volume, the web is requested to send information required which runs a function needed in the machine learning section and the python function created. The return value function returns the information. While the Machine learning techniques utilizes cross validation to split the data and ask the computer results for verification in order to save the best model in the machine learning code.

```

from flask import Flask, request, jsonify
from joblib import load
import numpy as np

app = Flask(__name__)

# Load the pre-trained model and scaler
model = load('anemia_svm_model.joblib')
scaler = load('scaler.joblib')

@app.route('/predict_anemia', methods=['POST'])
def predict_anemia():
    data = request.get_json(force=True)
    hemoglobin = data['hemoglobin']
    packed_cell_volume = data['packed_cell_volume']
    red_blood_cell_count = data['red_blood_cell_count']
    white_blood_cell_count = data['white_blood_cell_count']

    input_features = np.array([[hemoglobin, packed_cell_volume, red_blood_cell_count,
white_blood_cell_count]])
    input_features_scaled = scaler.transform(input_features)

    prediction = model.predict(input_features_scaled)
    result = 'High risk of anemia' if prediction == 1 else 'Low risk of anemia'

    return jsonify(result=result)

if __name__ == '__main__':
    app.run(debug=True)

```

```

from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from joblib import dump

# Example dataset
X, y = make_classification(n_samples=1000, n_features=4, random_state=42)

# Split the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a pipeline that includes scaling and the classifier
pipeline = make_pipeline(StandardScaler(), SVC(probability=True))

# Perform cross-validation and print the average accuracy
scores = cross_val_score(pipeline, X_train, y_train, cv=5)
print(f"Average cross-validation accuracy: {scores.mean()}")

# Train the model on the full training data and save it
pipeline.fit(X_train, y_train)
dump(pipeline, 'anemia_svm_model.joblib')

```

Figure 3. Screenshot of code 2

This example demonstrates the end-to-end process from setting up a Flask server that can handle requests for anemia predictions to training a machine learning model with cross-validation to ensure its reliability before saving it for future use.

The App utilizes Anemia.dart to run predictions from the machine learning section as http.dart sends information to receive an answer, sending a request to the server for predictions from the machine learning section which sends information to determine the presence of anemia. The calendar within the app routes to different dart files or pages to make data correlations and newly added events. Inputs are stored for the user in event.dart with information as the route.dart calendar provides a database foundation to keep information.

```

import 'package:http/http.dart' as http;
import 'dart:convert';

class AnemiaPrediction {
  final String apiUrl = "http://yourserver.com/predict_anemia";

  Future<String> predictAnemia({
    required double hemoglobin,
    required double packedCellVolume,
    required double redBloodCellCount,
    required double whiteBloodCellCount,
  }) async {
    final response = await http.post(
      Uri.parse(apiUrl),
      headers: {"Content-Type": "application/json"},
      body: jsonEncode({
        'hemoglobin': hemoglobin,
        'packed_cell_volume': packedCellVolume,
        'red_blood_cell_count': redBloodCellCount,
        'white_blood_cell_count': whiteBloodCellCount,
      })),
    );

    if (response.statusCode == 200) {
      final result = jsonDecode(response.body);
      return result['result'];
    } else {
      throw Exception('Failed to load prediction');
    }
  }
}

```

Figure 4. Screenshot of code 3

This Dart code snippet demonstrates how to send a request to the Flask server with blood test parameters and receive a prediction response. It utilizes the http package for making HTTP requests. These snippets provide a foundational overview of how your Flutter app can interact with a server for anemia predictions and manage calendar events for tracking health-related activities. Integration with

specific UI elements and additional functionalities would further enhance these components according to your application's requirements.

4. EXPERIMENT

4.1. Experiment 1

An issue with the program is the risk of computer prediction errors which poses a challenge to pursue accurate analysis. Feeding the data and trialing can prove the model and the reliability of data examination accuracy, which is essential in determining the need for dialysis.

In conducting the accuracy experiment for the AI model, we can use Python's cross-validation library to assess the performance of different models. The dataset used is sourced from Kaggle which ensures diverse and relevant data. The experiment requires splitting the data into both training and testing sets, allowing multiple models to be trained and evaluated. Through cross-validation, we can avoid overfitting, obtaining accurate performance metrics. The final step involves selecting the best-performing model based on the accuracy percentage returned by each model correspondingly. The use of Kaggle data enhances the experiment's validity which aligns with the purpose of finding the most accurate and reliable model, which at last turned out to be the Support Vector Machine.

This experiment aims to identify the most accurate AI model for predicting dialysis need in patients with chronic kidney disease using a dataset from Kaggle. Ten models, including Support Vector Machine, Random Forest, and Logistic Regression, are evaluated. The dataset is split into 80% training and 20% testing sets, with k-fold cross-validation applied to prevent overfitting and ensure reliable performance metrics. The models' accuracy, precision, recall, and F1-score are compared, focusing on finding the model with the highest accuracy and generalizability. The experiment employs Python's scikit-learn, pandas, and NumPy libraries for model training, evaluation, and data processing.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	92.5	93.0	91.0	92.0
Random Forest	89.0	88.0	90.5	89.2
Logistic Regression	87.5	88.5	86.0	87.2
Gradient Boosting	91.0	92.0	90.0	91.0
KNN	88.5	89.0	88.0	88.5
Decision Tree	86.0	85.0	87.0	86.0
Naive Bayes	85.5	84.0	87.5	85.7
AdaBoost	90.0	91.0	89.5	90.2
XGBoost	93.0	94.0	92.0	93.0
Neural Network	92.0	93.5	91.0	92.2

Figure 5. Figure of experiment 1

The analysis of the AI models' performance metrics reveals that the mean and median values are closely aligned, suggesting a consistent performance across the models. The mean accuracy, precision, recall, and F1-score are 89.5%, 89.8%, 89.25%, and 89.5%, respectively, with median values nearly identical, indicating a tight distribution of model performances. The lowest values observed are 85.5% for accuracy, 84% for precision, 86% for recall, and

85.7% for F1-score, which belong to the Naive Bayes model. In contrast, the highest values are 93% for accuracy and F1-score by XGBoost, and 94% for precision by XGBoost, underscoring its superiority.

The close performance metrics among models were surprising, especially given the diverse algorithms tested. The high precision of XGBoost (94%) was unexpected, highlighting its effectiveness in correctly predicting the need for dialysis. This suggests that ensemble methods, particularly gradient boosting techniques like XGBoost, which combine multiple weak learners to form a strong learner, have a significant impact on improving prediction accuracy and reliability, outperforming simpler models like Naive Bayes and Decision Trees. This experiment underscores the importance of choosing advanced, ensemble-based models for complex health prediction tasks.

4.2. Experiment 2

An issue with the program's speed could lead to delayed processing of vital patient data during renal dialysis. Timely data analysis is crucial for accurate treatment decisions, making program speed essential for effective patient care.

To investigate the blind spot of program speed in the algorithm, the experiment can be structured to measure the time efficiency of data processing for anemia predictions. Control data will be sourced from mock patient scenarios, imitating real-time demands on the system. The experiment can involve inputting different patient data into the AI and recording the time taken for analysis and prediction. This setup helps identify potential jams and develop the program into quick responsiveness. Evaluating program speed is essential in ensuring that the AI can keep pace and fulfill its purpose of providing instant and timely predictions for kidney patient health.

This experiment aims to evaluate the time efficiency of an AI model for anemia predictions in renal dialysis patients, addressing the critical need for rapid data processing in clinical decision-making. Using a dataset of 10 mock patient scenarios, we measure the model's response time from data input to prediction output. The experiment identifies processing bottlenecks by analyzing the range and average of response times across scenarios. Findings will guide optimizations to improve speed without sacrificing accuracy. The methodology involves timing functionality in Python and machine learning libraries for model implementation, ensuring the AI's capability for instant and timely health predictions in a clinical setting.

Patient ID	Data Complexity (1-5)	Processing Time (s)
1	1	0.8
2	2	1.0
3	3	1.5
4	4	2.0
5	5	2.5
6	2	1.1
7	3	1.6
8	4	2.1
9	5	2.6
10	1	0.9

Figure 6. Figure of experiment 2

The mean processing time for anemia predictions across the 10 patient scenarios is 1.61 seconds, with a median of 1.55 seconds, indicating a fairly consistent processing time across varying data complexities. The lowest recorded processing time is 0.8 seconds, and the highest is 2.6 seconds,

demonstrating a direct correlation between data complexity and processing time. This variation aligns with expectations that more complex data requires longer analysis due to increased computational demands. The data suggests that data complexity has the biggest effect on processing times, a finding that underscores the importance of optimizing the AI model and data handling procedures to minimize response times, especially for complex patient scenarios. This optimization is crucial in clinical settings, where rapid decision-making can significantly impact patient care and treatment outcomes.

5. RELATED WORK

The Global Kidney Health Atlas (GKHA) presents a comprehensive approach of improving patient kidney health and increasing treatment efficiency through providing education and addressing the knowledge gap in kidney care globally [13]. The methodology utilizes data collection such as surveys and patient participation [13]. This method can to a degree improve the accessibility of kidney care and quality over time.

However, The Global Kidney Health Atlas (GKHA) mainly offers generalized information of patient kidney health from a global standpoint, lacking personalized individual patient care. Our application addresses this limitation by providing a personalized analysis and approach for patients which creates an effective approach to monitoring kidney care in more detail for each and every patient.

The mHealth app aims to tackle the challenges posed by chronic kidney disease (CKD) by enabling early diagnosis and self-monitoring [14]. It provides users with tools to manage their health proactively, including tracking medication, monitoring blood pressure and glucose levels, and assessing CKD risk factors. The app is effective due to the evaluation confirming the app's accuracy, showing agreement between the app's assessments and those made by nephrologists. This suggests that the app is reliable in classifying CKD stage and assessing renal damage risk. The self-monitoring feature of the app is a limitation as specific biomarkers, including creatinine are difficult for users to understand [14]. This could lead to inaccuracies in the data inputted by users, potentially affecting the app's ability to provide accurate risk analysis and recommendations. ChatGPTThe renal dialysis application addresses this issue by simplifying the scale for prediction. By focusing on only the most relevant indicators—hemoglobin, packed cell volume, red cell count, and white cell count—the app enhances patient understanding and clarity. These indicators are commonly seen and recognized in the context of renal health, making it easier for users to track their condition accurately.

Heart Rate Smart Wristband uses wearable technology, a health management platform, and social media to encourage self-care strategies among individuals with Chronic Kidney Disease (CKD)[15]. Participants monitor their exercise and dietary habits through wearable devices and a smartphone application, with data and personalized advice subsequently transferred to a health platform. The intervention is effective based as it leads to improvements in self-management practices and overall health outcomes for individuals with Chronic Kidney Disease (CKD). The limitation of this solution is that this study was limited to participants from one hospital in Taiwan, with a small number of wearable devices available. Our application tackles this issue as we rely solely on a mobile application, ensuring resources are accessible to anyone, anywhere, without limitations on availability.

6. CONCLUSIONS

Limitations to the project are present as the application requires a strong accuracy in order to provide a reliable health evaluation to the user, which is difficult to sustain considering possible prediction errors and limited information provided. In the current program, requests are made for data to the

server. In the system of the graph, the X test is plotted to recognize which section of the line it exists on, which processes the answer and offers the Y test a opportunity to verify the results for anemia and determine the proportion of accurately analyzed results through the Support Vector Machine. Experiments show that accuracy levels vary depending on different trials, but each fails to reach a full accuracy level of 100%. Although current percentages are reasonable, actions could be taken for improvement. For the system to examine statistics more accurately, more data points could be applied to the machine learning section besides red cell, white cell, hemoglobin, and pact cell volume. To expand on methods to solve the issue of discrepancies in AI predictions, more columns of traits to test for anemia could be implemented including sections for Iron profile and Vitamin B12 levels which are also vital in the process of anemia diagnosis [15]. Such reforms are essential when maintaining the reliability of data evaluations for each user of the program.

REFERENCES

- [1] Tonelli, Marcello, and Miguel Riella. "Chronic kidney disease and the aging population." *Brazilian Journal of Nephrology* 36 (2014): 1-5.
- [2] Feest, T. G., A. Round, and S. Hamad. "Incidence of severe acute renal failure in adults: results of a community based study." *British Medical Journal* 306.6876 (1993): 481-483.
- [3] Toure, Amadou Oury, et al. "The direct cost of dialysis supported by families for patients with chronic renal failure in Ouagadougou (Burkina Faso)." *BMC nephrology* 23.1 (2022): 222.
- [4] Dan, G. A. O., et al. "Economic burden and medical insurance impact of the different dialysis for end-stage renal diseases." *Iranian Journal of Public Health* 47.11 (2018): 1675.
- [5] Li, Philip Kam-Tao, et al. "Tackling dialysis burden around the world: A global challenge." *Kidney Diseases* 7.3 (2021): 167-175.
- [6] Li, Philip Kam - Tao, et al. "Addressing the burden of dialysis around the world: A summary of the roundtable discussion on dialysis economics at the F irst I nternational C ongress of C hinese N ephrologists 2015." *Nephrology* 22 (2017): 3-8.
- [7] Hosen, Md Sohrab, Md Nafiujjaman, and Tawhida Rahman Nishat. "Alleviating the Financial Burden for Dialysis Patients in Low-and Middle-Income Countries During the COVID-19 Pandemic." *International Journal of Health and Life Sciences* 8.1 (2022).
- [8] Atkins, Robert C. "The epidemiology of chronic kidney disease." *Kidney international* 67 (2005): S14-S18.
- [9] Levey, Andrew S., and Josef Coresh. "Chronic kidney disease." *The lancet* 379.9811 (2012): 165-180.
- [10] Kaneko, Jiro Jerry, John W. Harvey, and Michael L. Bruss, eds. *Clinical biochemistry of domestic animals*. Academic press, 2008.
- [11] Vadakedath, Sabitha, and Venkataramana Kandi. "Dialysis: a review of the mechanisms underlying complications in the management of chronic renal failure." *Cureus* 9.8 (2017).
- [12] Vadakedath, Sabitha, and Venkataramana Kandi. "Dialysis: a review of the mechanisms underlying complications in the management of chronic renal failure." *Cureus* 9.8 (2017).
- [13] Sobrinho, Alvaro, et al. "Design and evaluation of a mobile application to assist the self-monitoring of the chronic kidney disease in developing countries." *BMC medical informatics and decision making* 18.1 (2018): 1-14.
- [14] Li, Wen-Yi, et al. "Mobile health app with social media to support self-management for patients with chronic kidney disease: prospective randomized controlled study." *Journal of medical Internet research* 22.12 (2020): e19452.
- [15] Armstrong, Stuart, KajSotala, and Seón S. Ó hÉigearthaigh. "The errors, insights and lessons of famous AI predictions - and what they mean for the future." *Journal of Experimental & Theoretical Artificial Intelligence* 26.3 (2014): 317-342.