International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 38, No. 4, August, 2024, pp. 1305-1318

Journal homepage: http://iieta.org/journals/ria

Enhanced K-Nearest Neighbors for Smart Cardiovascular Disease Prediction in IoT System

Farida Brahimi[1*], Aicha Aid[1], Mourad Amad[1], Abdelghani Mehennaoui[1], Abderahmane Baadache[2]

[1] LIM Laboratory, Department of Computer Science, Faculty of Sciences and Applied Sciences, University of Bouira, Bouira 10000, Algeria
[2] Department of Computer Sciences, University of Algiers 1, Algiers 160000, Algeria

Corresponding Author Email: f.brahimi@univ-bouira.dz

https://doi.org/10.18280/ria.380424

Received: 25 February 2024
Revised: 1 May 2024
Accepted: 6 June 2024
Available online: 23 August 2024

Keywords:
cardiovascular disease, classification model, internet of things, machine learning, K-NN classifier

ABSTRACT

Cardiovascular disease (CVD) is becoming more prevalent as a health issue and ranks as a top cause of mortality globally. Effectively identifying CVD is frequently a challenging process, given that minor errors can result in significant consequences. To address this challenge, healthcare organizations have recently embraced Internet of Things (IoT) to collect patients' vital signs using wearable sensors, these data are then stored and transmitted to machine learning (ML) based prediction systems. Among the various ML algorithms, the K-Nearest Neighbors (K-NN) algorithm stands out for its simplicity and effectiveness in CVD prediction. However, its reliance on majority voting can lead to classification errors, especially when test vectors are closer to minority class neighbors. To address this limitation, we propose the Enhanced K-Nearest Neighbors (E-KNN) algorithm, specifically designed to refine classification accuracy by incorporating a weighted distance measure that considers both neighbor proximity and class distributions. The E-KNN model has undergone comprehensive testing in comparison to standard ML methods. The experimental findings demonstrate that the introduced model surpasses current methodologies based on performance assessment indicators, recording a 91.43% notable accuracy level. To leverage the E-KNN algorithm, we have developed an IoT platform that gathers crucial patient data and transmits it to the E-KNN based model.

1. INTRODUCTION

Good health is essential and represents a fundamental desire of every individual, necessary to enable them to efficiently fulfill their daily responsibilities and achieve their long-term goals. However, the healthcare industry is currently facing several challenges, such as a shortage of medical staff, an escalating healthcare costs, and an aging population. In 2000, the world population over 60 years of age was 11%, and this percentage is projected to increase to 22% by 2050 [1]. These challenges are compounded by lifestyle factors such as unhealthy diets, smoking, obesity, and stress, which contribute to an increased prevalence of chronic diseases, notably CVD.

CVD encompass a range of conditions impacting the heart and the vascular system. As per the World Health Organization (WHO), between 1990 and 2019, the global patient count for CVD surged from 271 million to 523 million, while fatalities attributed to these conditions climbed from 12.1 million to 18.6 million, constituting 32% of worldwide deaths in 2019 [2]. In Algeria, the annual death rate due to CVD is projected at 34% [3].

Diagnosing CVD is an incredibly complex task, and multiple tests are typically necessary to reach a precise conclusion. If a heart disease goes undetected and untreated, many complications can arise, such as arrhythmias peripheral artery disease, and sudden cardiac arrest, among others. Fortunately, with the emergence of artificial intelligence and IoT, it is now possible to predict CVD at an early stage, which can contribute to mitigating complications and reducing mortality rates.

The concept of "IoT" encompasses a network of physical items embedded with sensors, software, and various technological features. These objects are created to connect with other objects and systems over the internet, making it easier for them to share data [4]. These objects can be simple household appliances, wearable devices, or highly complex industrial equipment [5]. The IoT is a vital technological tool in the healthcare sector, it facilitates the real-time detection, tracking, and monitoring of vital signs, including electrocardiogram (ECG or EKG), blood glucose levels, respiratory rate, and blood lipid levels, thereby aiding in the detection and prevention of various illnesses [6, 7]. The use of ML for analyzing this data is rapidly increasing, contributing to the reduction of healthcare costs and the improvement of the patient-doctor relationship [8]. The concept of "ML" was introduced by Arthur Samuel in 1959, described as a discipline allowing computers to acquire knowledge autonomously without being explicitly programmed [9]. ML harnesses data and algorithms to emulate human learning, aiming for continuous improvement in accuracy. It employs techniques and tools to uncover patterns within datasets, building models that faithfully represent the data. These models enhance our grasp of phenomena, such as pinpointing risk factors for CVD, and forecasting future occurrences, like identifying individuals at elevated risk for CVD. When effectively implemented, ML empowers healthcare professionals to make precise diagnoses, select optimal treatments, and reduce medical expenses.

K-NN is fundamentally a non-parametric classification algorithm where the decision on the class of a new observation is based on the majority of classes of its K nearest neighbors. Although simple and intuitive, standard K-NN can be ineffective or inaccurate in situations where data from minority classes are physically close to the new observation, a common scenario in unbalanced medical datasets.

To overcome this drawback, we propose the Enhanced K-Nearest Neighbors (E-KNN) algorithm, which enhances classification accuracy by taking into account both neighbor proximity and class distributions. This refinement of the traditional majority voting method employs a weighted distance measure that incorporates not only neighbor closeness but also their class distributions. The main modifications of the E-KNN algorithm are as follows:

- Neighbor weighting based on class density: E-KNN introduces a probability factor P that weights the contribution of each neighbor based on the density of its class in the immediate neighborhood. This weighting allows for better consideration of minority classes, thus improving the algorithm's ability to handle imbalanced class distributions.
- Calculation and adjustment of new distances D_n (X,Y): The distances between each test point X and its K neighbors Y, D_n(X,Y), are calculated using a weighting factor P and a distance adjustment parameter d. The parameter d adjusts the measured distance between points to avoid unfairly favoring either the majority or minority classes. This ensures fair treatment of all classes and improves the overall accuracy of the classification.
- Selection of the most representative neighbor: By sorting the newly calculated distances D_n(X,Y) in ascending order, the algorithm selects the most influential neighbor for classifying the test vector X. This neighbor is chosen not only for its proximity but also for its statistical relevance, ensuring a more precise and balanced classification.

This paper introduces E-KNN as a novel approach to CVD prediction, combining IoT capabilities with advanced ML techniques to create a more accurate, responsive, and patient-centric predictive model. Through comprehensive testing and analysis, E-KNN demonstrates superior performance over standard ML methods, showcasing its potential to significantly improve the diagnosis, prediction, and management of CVD in an IoT-enabled healthcare landscape.

The contributions of this work are twofold. Firstly, we introduce an enhanced version of the K-NN algorithm, denoted as E-KNN, which serves as the foundation for a CVD prediction system. We thoroughly evaluate its performance, comparing it to traditional ML algorithms such as K-NN, Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Support Vector Machine (SVM), and other existing research works in the field. Secondly, we developed a biomedical data acquisition system, comprising an Arduino board and multiple device sensors: a blood pressure sensor for measuring arterial pressure, a heart rate sensor for pulse monitoring, and an AD8232 electrocardiographic sensor for recording cardiac electrical activity. These devices collect physiological data which are then transmitted and processed by the E-KNN algorithm to enhance diagnostic precision for cardiovascular conditions.

The subsequent sections of this document are organized in the following manner: Section 2 describes ML algorithms used in this study and related works on CVD prediction; Section 3 outlines the architecture of the CVD prediction model and describes the proposed E-KNN algorithm. Section 4 outlines the experimental framework and discusses the outcomes derived from assessing the models. In conclusion, Section 5 offers insights and explores prospective future endeavors aimed at expanding and refining the suggested CVD prediction model.

2. LITERATURE REVIEW

2.1 Used ML classifiers

In the realm of ML, three principle classes of algorithms exist: supervised, unsupervised, and reinforcement learning as illustrated in Figure 1.

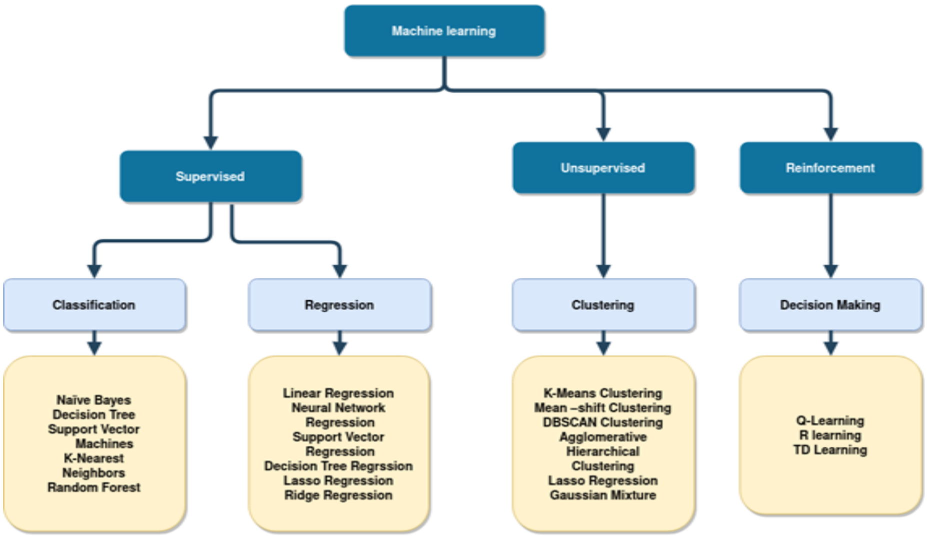


Figure 1. Types of ML algorithms [10]

In this paper, we focus on supervised learning, which involves labeled data based on the desired outcome. This form of learning is frequently utilized for predictive analysis. On the other hand, unsupervised learning addresses problems of grouping, association, and dimension reduction by building models from unlabeled data. Reinforcement learning uses learning algorithms that learn from repeated experiences through trial and error. This technique has been successfully applied to various problems, such as robotic control, task scheduling, and telecommunications.

The study discussed in this paper used various supervised machine learning algorithms, including K-NN [11], SVM [12, 13], DT [14, 15], RF [16, 17], and LR [18]. In the following, a comprehensive description of the K-NN algorithm will be provided, coupled with a concise overview of other algorithms such as SVM, DT, RF and LR.

KNN: The K-NN algorithm is a supervised classification algorithm that allocates a class to a test vector by comparing it to a set of labeled vectors recorded during the learning phase. This comparison aims to extract the K vectors that are closest to the considered vector in terms of distances [19]. There are several formulas to calculate the distance between two vectors $X(x_1, x_2, \ldots, x_n)$ and $Y(y_1, y_2, \ldots, y_n)$, with the most commonly used being Euclidean distance, which is delineated by the equation below, Eq. (1)

$$D(X,Y) = \sqrt{\sum_{i=1}^{i=n} (xi - yi)^2} \qquad (1)$$

The class assigned to the test vector is the most voted class among the k classes obtained in the comparison step [20].

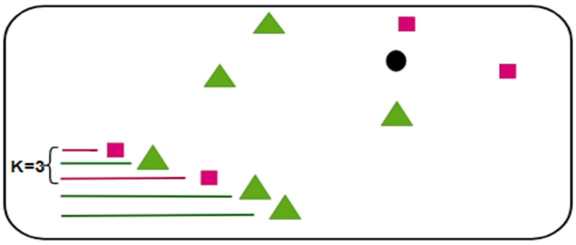


Figure 2. Class prediction of tested vector by K-NN

Figure 2 is an illustrative example of classification by K-NN. The K-NN algorithm calculates the distances between the data that we want to predict the class of (the black circle) and all the existing training data (the pink squares and green triangles). These distances are then sorted from smallest to largest. A number of neighbors, K, is chosen (for example, k=3), and the majority class (the red squares) is assigned to the tested data. The pseudo code of the KNN is as follows:

```
Input: Training dataset, Test dataset, K
Output: Class of test vector X
Begin
For each test vector X in the Test dataset
    For each training vector Y in the Training dataset
        Calculate the distance between X and Y, denoted as D(X, Y), using Eq. (1).
    End For
    Sort all calculated distances D(X, Y) in increasing order.
    Select the K smallest distances to determine the K closest neighbors.
    For each of the K neighbors
        Count the frequency of each class among these neighbors.
    End For
    Assign the test vector X to the class most frequent among the K closest neighbors.
End For
End
```

SVM: The SVM algorithm is widely used for classification and regression analysis, especially for binary classification problems. SVM functions by projecting the input data into a higher-dimensional feature space, where it identifies the best hyperplane that enlarges the gap between the two categories. This hyperplane is defined by the support vectors, which represent the data points nearest to the decision margin. By increasing the separation between the hyperplane and the nearest data points of each category, SVM delivers a strong classification framework that demonstrates reduced sensitivity to noise and outliers.

LR: Logistic regression operates as a predictive model for categorizing a dependent variable Y by employing a sigmoid function on multiple independent variables X_i. This model is refined through gradient descent to ascertain the best weights for X_i that reduce the logistic loss function, thereby forecasting the class with the greatest probability estimation.

DT: The DT is an easy-to-understand decision-making tool that uses a tree-like graph to make decisions. It selects the best predictor feature by calculating a splitting criterion, which is used to divide the data into subsets until a stopping criterion is met. The stopping criterion could be a maximum tree depth, a minimum number of instances in a node, or a threshold value for the splitting criterion. Pruning may be used to remove complex or overfitted branches. To make a prediction, input data is fed into the tree, and the algorithm follows the tree's branches based on the input features until it reaches a leaf node containing the predicted outcome.

RF: The RF algorithm functions by generating a collection of decision trees, where each tree is built from a randomly chosen subset of training data. This method of selecting subsets at random aids in mitigating overfitting risks and increases the variety among the trees. Aggregating the outcomes from all trees, the RF algorithm yields predictions that are both more precise and robust than those from an individual decision tree. For classification tasks, the algorithm aggregates the predictions of all the individual trees and selects the most frequent prediction as the final output. This approach not only improves the precision of the forecast but also aids in reducing the effects of noisy or outlier data points. For regression tasks, Random Forest Algorithm takes the average prediction of all the trees as the final output, which provides a smooth and continuous prediction surface that can handle nonlinear relationships between the independent and dependent variables.

2.2 Survey of previous work

After describing the ML algorithms we have implemented for CVD prediction, we now move on to explore some relevant studies conducted by other researchers in the same field.

Rajdhan et al. [21] looked at how well the DT, LR, RF, and NB algorithms could predict CVD using the dataset provided by the UCI ML Repository. This study revealed that the RF algorithm achieved the top accuracy rate of 90.16%, establishing it as the most effective method for heart disease prediction.

Ware et al. [22] conducted a study to compare six different machine learning techniques for heart disease prediction using the Cleveland dataset. The dataset was preprocessed by removing all noisy and missing data before analysis. The techniques evaluated were SVM, K-NN, RF, DT, LR, and NB, employing a range of performance measures. The findings indicated that SVM achieved the greatest accuracy rate of 89.34%, surpassing the performance of other methods.

Magar et al. [23] developed a ML-based web application for heart disease prediction. This research utilized various ML algorithmes, such as DT, LR, NB, and SVM. To train these algorithms, the authors allocated 75% of the Cleveland dataset, reserving the final 25% for evaluating their precision. The outcomes revealed LR to be the top-performing algorithm, achieving an 82.89% accuracy rate. SVM followed closely with an accuracy of 81.57%, whereas both DT and NB recorded accuracy rates of 80.43%.

Shah et al. [24] developed a model for predicting heart disease by applying ML methods such as RF, DT, K-NN, and NB. They trained these classifiers with the Cleveland dataset, which was pre-processed before being utilized in the model. The findings indicated that K-NN achieved the top accuracy rate of 90.78%.

Arghandabi and Shams [25] created a predictive model for heart disease employing a variety of ML classifiers, such as DT, K-NN, Gradient Boosting (GB), SVM, and LR algorithms. The study used 73% of the UCI heart dataset for training and 37% for testing the accuracy of the algorithms. The outcomes demonstrated that K-NN recorded a notable accuracy of 85.7%.

Reddy et al. [26] developed a system capable of diagnosing heart disease with the help of ten ML classifiers, including NB, LR, SMO, IBK, AdaBoostM1 paired with Decision Stump, AdaBoostM1 paired with LR, Bagging paired with REPTree, Bagging paired with LR, JRip, and RF. These classifiers underwent training utilizing the comprehensive attributes from the Cleveland heart dataset and optimal attribute sets derived from three evaluators: correlation-based feature subset, chi-squared attribute, and ReliefF attribute. The findings indicated that Sequential Minimal Optimization, when applied to the complete attribute set, reached an accuracy of 85.148%. Meanwhile, the most precise results, with an accuracy of 86.468%, were achieved with the optimal attribute set identified by the chi-squared attribute evaluator.

Kavitha et al. [27] crafted a combined model for forecasting heart disease by integrating RF and DT classifiers. This model attained an 88.7% accuracy rate in predicting heart disease with the Cleveland dataset. Experimental findings suggested that this integrated model outperformed the individual RF and DT models in terms of efficacy.

Chowdhury et al. [28] examined the effectiveness of DT, LR, K-NN, NB, and SVM as ML algorithms for the early detection of CVD. They compiled a dataset consisting of 564 cases and 18 attributes from healthcare sectors and hospitals in Sylhet, Bangladesh. The findings revealed that SVM achieved the top accuracy rate of 91% for the selected instances of the dataset.

Menaa et al. [29] focused on developing and enhancing a Dense-DNN based model for predicting heart disease. They assessed the Dense-DNN model's performance in comparison with various ML models, such as SVM, LR, RF, DT, GNB, K-NN, and XGBoost. To refine the model's efficiency, the researchers applied a genetic algorithm for choosing the most pertinent attributes. Without feature selection, the Dense-DNN model reached a 91.7% accuracy level, which increased to 95% when feature selection was implemented. The model exhibited superior accuracy with MAE/RMSE values of 0.083/0.289 without attribute selection and 0.050/0.224 with attribute selection.

Pan et al. [30] carried out a comprehensive examination of numerical, categorical, and mixed features using cutting-edge ML techniques. The research employed a range of ML algorithms, such as Gradient Boosting (GB), Extreme Gradient Boosting (XGBoost), AdaBoost (AdaB), CatBoost (CatB), artificial neural networks (ANN), RF, SVM, DT, and LR. For their analysis, the authors selected the widely recognized Cleveland heart disease dataset as a benchmark for their research. The evaluation of performance metrics indicated that categorical features surpassed numerical and combined features in effectiveness. Additionally, the study revealed that a combination of SVM and AdaBoost classifiers, when applied to categorical features, yielded the best results for predicting CVD.

Almulihi et al. [31] introduced a deep-stacking ensemble approach aimed at the early detection of cardiac conditions using basic data and symptoms. Along with SVM as a learning meta-model, the model includes two hybrid models that have already been optimized and trained: the Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) and the Convolutional Neural Network-Recurrent Grid Unit (CNN-GRU). The Recursive Feature Elimination (RFE) selection technique was used to select the most relevant attributes from two datasets: Cleveland and Heart Diseases. The model in question was assessed against other machine learning models, showing that its performance greatly outshined those implemented in the research.

Jansi Rani et al. [32] aimed to develop a wearable biomedical prototype for predicting the occurrence of cardiac conditions. The prototype included an ECG sensor to monitor the variation in ECG patterns, and several algorithms were trained using the Cleveland dataset to detect heart conditions at an early stage. Findings indicated that the Random Forest algorithm achieved the top accuracy rate of 88% in forecasting heart conditions.

Umer et al. [33] introduced an intelligent healthcare system that leverages IoT and Cloud technologies. This system incorporates a deep learning CNN model to classify heart failure patients into two categories: alive or deceased. To continuously monitor the health status of cardiac patients in real-time, a set of sensors tracks various vital signs, including Heart Rate (HR), Blood Pressure (BP), Temperature, Blood Glucose, Cholesterol, and Electrocardiogram (ECG) signals. These sensor data are transmitted to a Cloud web server for processing and subsequently forwarded to the CNN model for predicting the patient's health condition. The dataset used for this study contains 13 attributes and was sourced from the UCI repository called Heart Failure Clinical Records. The predictive model developed in this research attains an impressive accuracy rate of 92.89%.

Subahi et al. [34] the primary aim of this investigation is to enhance the precision of heart disease assessments through the utilization of the Modified Self-Adaptive Bayesian algorithm (MSABA). Sensors are deployed to monitor a range of cardiac parameters, including ECG pulses, temperature, heart rate, blood glucose, lipid levels, and other relevant factors, to continuously observe the overall health status of individuals afflicted with heart conditions. The model's training and testing phases employed datasets from Cleveland, Hungarian, and a merged dataset known as CH. The proposed approach, MSABA, demonstrates an impressive accuracy rate of 90%.

Djerioui et al. [35] proposed a model using the SVM algorithm for the effective prediction of heart disease. The authors employed Neighborhood Component Analysis (NCA) to select the most relevant attributes in order to enhance the performance of the suggested method. The proposed model achieved an accuracy rate of 85.43%.

3. PROPOSED RESEARCH METHODOLOGY

Technological advancements such as IoT, ML, and deep learning have a significant impact on the healthcare sector. These innovations make it possible to continuously monitor a patient's health condition in real-time, which in turn allows for the prompt identification of potential health concerns. To provide more accurate assessments of cardiac illnesses, we have developed an IoT platform that leverages biomedical sensors to collect essential medical data. Subsequently, this data is filtered, processed, and directed to a CVD prediction system based on the E-KNN algorithm. In Figure 3, the depicted experimental process outlines the functioning of the envisaged predictive system for CVD. The system comprises two modules: data acquisition module and predictive analysis module.

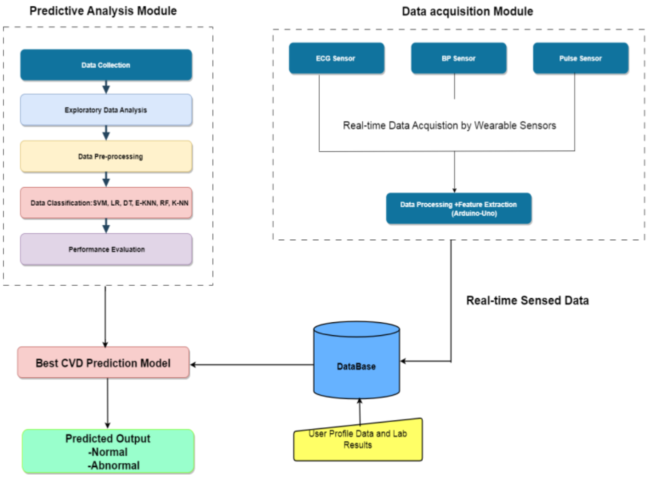

Predictive Analysis Module
Data Collection
Exploratory Data Analysis
Data Pre-processing
Data Classification:SVM, LR, DT, E-KNN, RF, K-NN
Performance Evaluation
Data acquisition Module
ECG Sensor
BP Sensor
Pulse Sensor
Real-time Data Acquistion by Wearable Sensors
Data Processing +Feature Extraction (Arduino-Uno)
Real-time Sensed Data
DataBase
Best CVD Prediction Model
Predicted Output
-Normal
-Abnormal
User Profile Data and Lab Results

Figure 3. Architecture of the proposed system

3.1 Data acquisition module

This module is responsible for collecting and analyzing real-time data necessary for the diagnosis of CVD from wearable biomedical devices and a user profile, preparing it for use in ML predictions.

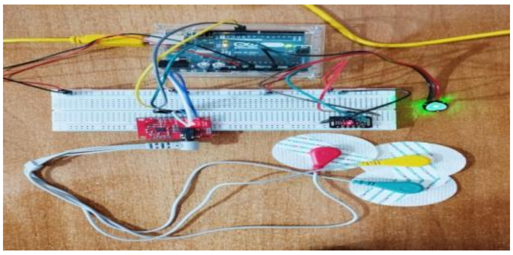


Figure 4. Developed hardware system

The hardware part of this data acquisition module consists of an Arduino board that acts as the brain of the system and multiple biomedical sensors, as depicted in Figure 4. Its primary role involves sensing and transmitting data via a serial communication connection. On the other hand, the software part comprises a Python application that receives the transmitted data, process it to extract usable features for ML modeling, and then store it in both a MySQL database and a CSV file.

The data collection operation starts by capturing the dataset features thalach, TRESTBPS, Slope, Oldpeak, and restecg from available sensors when the relevant button on the Python application is clicked.

The patient's continuous pulse rate is obtained using the pulse sensor. Upon clicking the Pulse Sensor button, 10 successive pulse rates are captured and sent to the application over a serial connection port. The highest value among these 10 rates is considered as the THALACH value. When the Blood Pressure Sensor button is pressed, the resting blood pressure TRESTBPS is measured and also transmitted to the application.

The AD8232 ECG sensor is used for capturing small electrical signals from the patient's heart with a sampling rate of 1000Hz. When the ECG Sensor button is pressed in the Python application, 60000 consecutives ECG data samples are transferred to the application and saved in a separate CSV file. This saved ECG signal is further processed to extract desired features like SLOPE, OLDPEAK, and RESTECG, as illustrated in Figure 5.

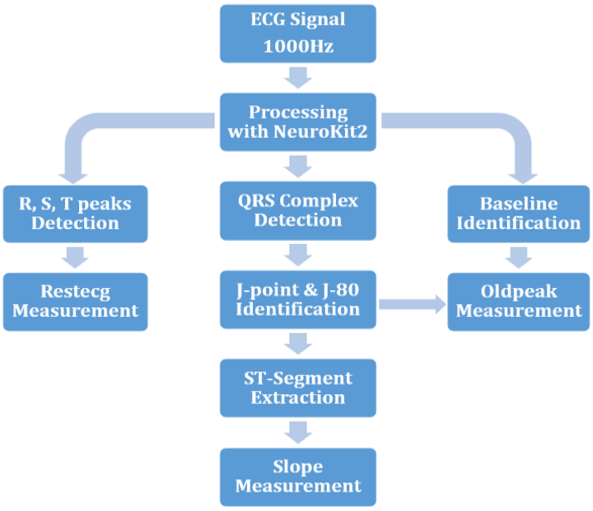


Figure 5. ECG signal processing steps

The ECG signal processing starts by detecting the QRS complex using the Pan-Tompkins algorithm [36]. It works by bandpass filtering the signal, differentiating it to enhance QRS complex peaks, squaring the signal to further emphasize these peaks, and then applying a moving window integration process to smooth the signal and detect QRS complexes as local maxima. This algorithm is implemented in the Python toolbox for neurophysiological signal processing NeuroKit2 [37].

Next, the ST-segment is extracted from the identified QRS complex and R-peaks. The ST-segment is an isoelectric section of the ECG signal that follows the QRS complex and precedes the T-wave. It provides information about the inclination and the steepness of the signal waveform. It is defined as a segment of the signal following the J-point (end of the QRS complex) and extending for an 80-120ms duration.

The ST segment is measured at a point 60-80ms after the J-point (J-60 and J-80) to avoid the influence of the J-point itself.

After the ST segmentation step, the SLOPE of the ST-segment is calculated. The slope of the ST-segment refers to the rate of change of the ST segment's amplitude over time. The amplitude of the ST-segment is defined as the deviation from the baseline. The slope is measured as the change in voltage between the J-point and the J-60/J-80 divided by the change in time. It is expressed in millivolts per millisecond (mV/ms). A positive slope indicates an upsloping or ascending ST-segment, while a negative slope indicates a downsloping or descending ST-segment.

The OLDPEAK in an ECG signal refers to the ST-segment depression (ST depression) induced by angina relative to rest. It is measured as the vertical distance (in millimeters) between the baseline and the J-point (or the J-60/J-80 points during an exercise stress test). On the other hand, the RESTECG categorical value is deduced based on resting electrocardiographic results such as T-wave inversion, oldpeak, and left ventricular hypertrophy (LVH).

Due to the unavailability of sensors to measure chest pain (CP), the number of major vessels colored by fluoroscopy (CA), serum cholesterol (CHOL), glucose level (FBS), and exercise-induced angina (EXANG), their lab values are directly stored in a personal profile along with the patient's age and sex.

The final extracted and computed values will serve as input features for the selected best model identified by the predictive analysis module. The model will leverage these features to make informed predictions based on the individual's health data, aiding in early detection and prevention strategies for CVD.

The system architecture of the data acquisition module is shown in Figure 6.

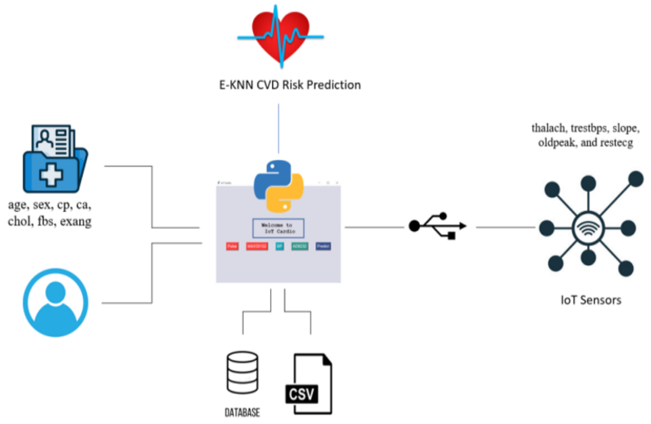

E-KNN CVD Risk Prediction
thalach, trestbps, slope, oldpeak, and restecg
age, sex, cp, ca, chol, fbs, exang
Welcome to IoT Cardio
IoT Sensors
DATABASE
CSV

Figure 6. System architecture of the data acquisition module

Predictive analysis module

The predictive analysis module consists of five main steps: data collection, exploratory data analysis, data preprocessing, data classification, and performance evaluation.

3.2.1 Data collection

The data collection step aims to gather relevant information from reliable sources such as medical records, surveys, genetic tests, and data sensors. For our study, we use the Cleveland Heart Disease dataset [38], a publicly available and well-known dataset. This dataset comprises medical data related to patients referred to the Cleveland Clinic Foundation between 1988 and 1990 for suspected heart disease. It contains 303 instances, with 13 independent variables representing clinical measurements and demographic information and one dependent variable as the target variable. The target variable has two classes. In class 1, heart disease is detected, whereas in class 0, there is no presence of heart disease. Table 1 furnishes an in-depth exposition of the Cleveland dataset.

Table 1. Cleveland heart dataset attributes

S. No.	Attribute	Description of attribute
1	AGE	Years of age (29-77)
2	SEX	0: male, 1:female
3	TRESTBPS	Standing blood pressure of the patient (in mm Hg) (94 to 200)
4	CHOL	Serum cholesterol (in mg/dl) (126 to 564)
5	CP	Type of chest pain 1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: nosymptoms
6	FBS	If fasting blood glucose >120 mg / dl 1: true, 0: false
7	RESTECG	Electrocardiographic result at rest 0 = normal, 1 = ST-T wave abnormality, 2 = definite left ventricular hypertrophy by Estes' criteria
8	THALACH	The maximum heart rate of the individual (71- 202)
9	EXANG	Angina induced by exercise 1 = yes, 0 = no
10	OLDPEAK	ST depression induced by exercise compared to rest (0 to 6.2)
11	SLOPE	the slope of the peak exercise ST segment 1 = up-sloping, 2 = flat, 3 = down-sloping
12	CA	Number of major vessels colored by fluoroscopy (0-3)
13	THAL	A blood disorder called thalassemia 3 = normal, 6 = fixed defect, 7 = reversible defect
14	Target	1: presence of heart disease, 0: absence of heart disease

3.2.2 Exploratory data analysis

Exploratory data analysis involves examining datasets to identify their main attributes, unveil relationships between variables, detect outliers and anomalies, and test underlying assumptions. Table 2 presents the statistical distribution of the different attributes of our dataset, such as minimum, maximum, average, standard deviation (STD), and missing values. According to our exploratory analysis of the data, we found no missing values in the Cleveland dataset.

Figure 7 shows the visualization graphs of the target class. It can be noted that there are 138 individuals without heart disease, representing 45.5% of the sample, while 165 individuals have heart disease, accounting for 54.5%. It indicates that the percentage of individuals with and without heart disease is almost equal, i.e., the dataset is balanced. A balanced dataset is crucial for attaining optimal classification results as it enables the model to be trained on an equal number of positive and negative instances.

Figure 8 illustrates the univariate analysis for the attributes of the dataset. It reveals that the dataset comprises eight categorical attributes (SEX, CP, FBS, RESTECG, EXANG, SLOPE, CA, and THAL) and five numerical attributes (AGE, trestbps, chol, thalach, oldpeak). Specifically:

Table 2. Statistical distribution of Cleveland dataset

Attribute	Mini	Max	Mean	STD	Missing Values
age	29.0	77.0	54.366337	9.082101	0
sex	0.0	1.0	0.683168	0.466011	0
cp	0.0	3.0	0.966997	1.032052	0
trestbps	94.0	200.0	131.623762	17.538143	0
chol	126.0	564.0	246.264026	51.830751	0
fbs	0.0	1.0	0.148515	0.356198	0
restecg	0.0	2.0	0.528053	0.525860	0
thalach	71.0	202.0	149.646865	22.905161	0
exang	0.0	1.0	0.326733	0.469794	0
oldpeak	0.0	6.2	1.039604	1.161075	0
slope	0.0	2.0	1.399340	0.616226	0
ca	0.0	4.0	0.729373	1.022606	0
thal	0.0	3.0	2.313531	0.612277	0
target	0.0	1.0	0.544554	0.498835	0

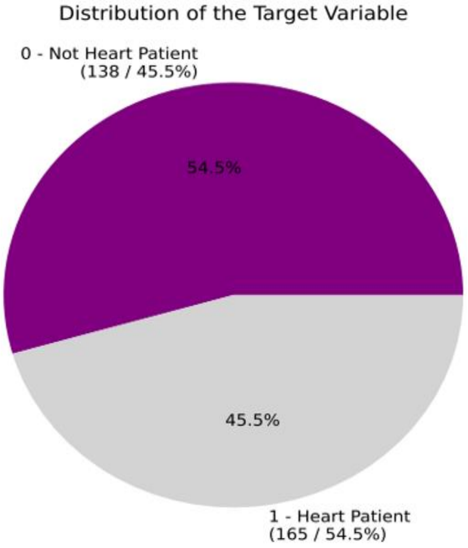


Figure 7. Target visualization

- SEX: There are more male participants than female.
- CP (Chest Pain Type): The distribution shows that type 0 (typical angina) is the most common, followed by non-anginal pain, atypical angina, and asymptomatic types.
- FBS (Fasting Blood Sugar): The majority of participants have fasting blood sugar below 120 mg/dl.
- RESTECG (Resting Electrocardiographic Results): Shows a mix, with the majority having type 1 (ST-T wave abnormality).
- EXANG (Exercise Induced Angina): Most participants did not experience angina induced by exercise.
- SLOPE: The slope of the peak exercise ST segment has a majority in slope 2.
- CA (Number of Major Vessels Colored by Fluoroscopy): A large number of participants have 0 major vessels colored by fluoroscopy, indicating no major blockages.
- THAL: The most common value is 2 (fixed defect), followed by 3 (reversible defect) and 1 (normal).
- AGE: The age distribution is roughly normal, centered around mid-50s.
- Trestbps (Resting Blood Pressure): Shows a normal distribution with a mean around 130 mm Hg.
- Chol (Serum Cholesterol): The distribution is slightly right-skewed, indicating a few participants with very high cholesterol levels.
- Thalach (Maximum Heart Rate Achieved): The distribution is slightly left-skewed, with most participants achieving a high maximum heart rate.

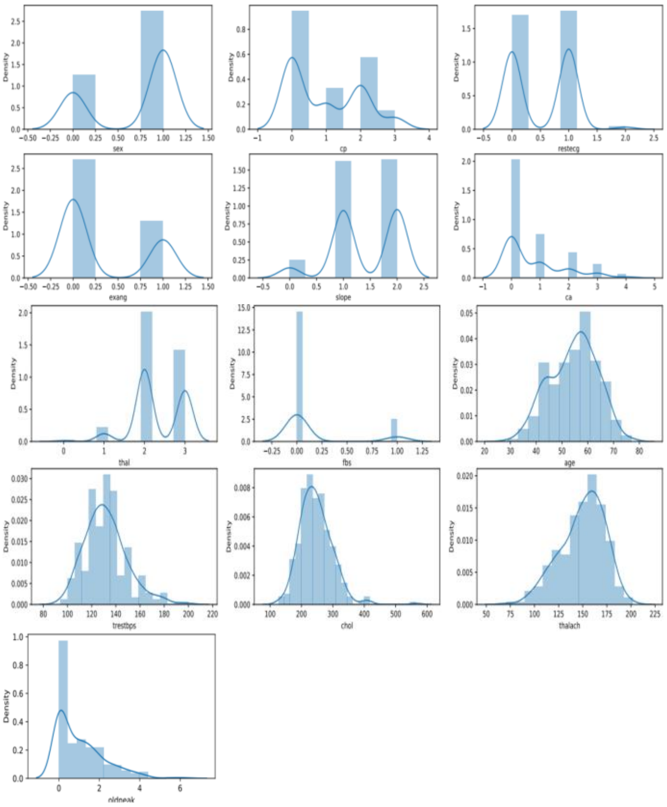


Figure 8. Univariate analysis of Cleveland attributes

The correlation matrix stands as a fundamental instrument in the conduct of exploratory data analysis, it helps us to select the most important variables for analysis. It is represented in the form of a table that displays Pearson correlation coefficients (P) between several variables. These coefficients gauge both the magnitude and orientation of the linear association between two variables, encompassing a numerical scale that spans from -1 to 1." If the correlation coefficient between two variables is greater than 0.7, we can conclude that these variables are strongly correlated [39]. When two variables are strongly correlated, it may suggest that they contain similar information, and one of them can be removed without affecting the analysis results.

By examining the correlation matrix presented in Figure 9, we observe that our variables are not strongly correlated and can be used in the development of the ML model.

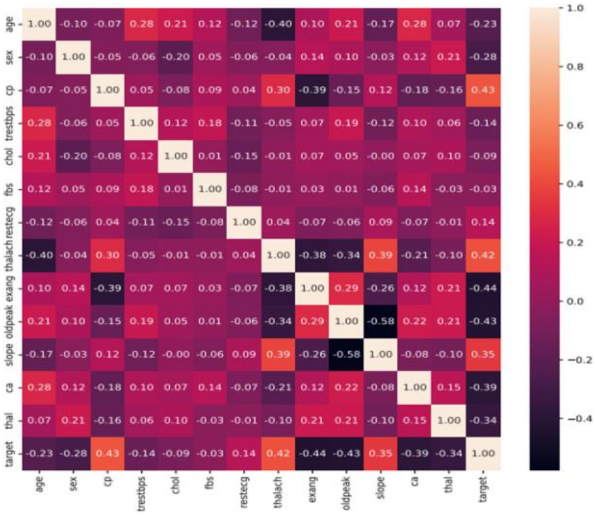


Figure 9. Correlation matrix

3.2.3 Data preprocessing

Data preprocessing is an essential phase in the development of reliable and precise ML models. It guarantees that the data is purified, uniform, and prepared for training. To preprocess our data, we followed two steps: removing duplicate records and normalizing the data using the StandardScaler technique [40]. This technique transforms each variable X so that its mean (μ_X) is zero and its standard deviation (σ_X) is equal to 1, using the Eq. (2)

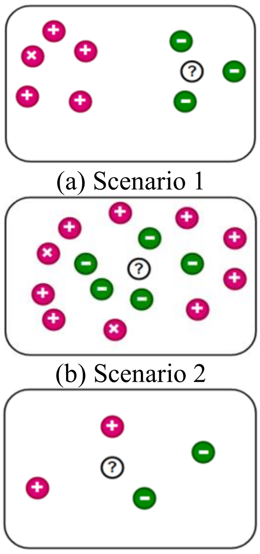
$$Xscaled = \frac{X - \mu x}{\sigma x} \tag{2}$$

3.2.4 Data classification

Data classification involves the systematic arrangement and tagging of data into specific groups, facilitating the discovery of patterns and insights for predictive purposes. In our methodology, the preprocessed dataset was segmented into two portions: 77% allocated for training and 23% designated for testing purposes. The training portion served to refine our developed model, E-KNN, alongside a suite of supervised ML techniques such as K-NN, SVM, DT, LR, and RF, aimed at segregating patients according to their heart disease risk levels. The effectiveness of these models was then assessed using the testing portion. To enhance the performance of our models, we fine-tuned the model's hyper parameters by harnessing the power of GridSearchCV with a cross-validation set defined at 5 folds.

Now, let's delve into the details of the proposed E-KNN algorithm, which we will use for classifying patients into categories with and without heart disease.

Proposed algorithm E-KNN. The traditional K-NN algorithm classifies a test vector into the most frequent category among its K nearest neighbors, which can sometimes lead to incorrect classifications, especially in cases where the test vector is physically closer to the elements of a minority category, as illustrated in Figure 10 which displays scenarios that were misclassified by the KNN algorithm.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Figure 10. Examples of misclassified data by K-NN

In the first scenario, the test vector is closer to the minority (negative) class. However, for K values greater than or equal to 7, the K-NN algorithm assigns the unclassified test vector to the majority (positive) class, even though it belongs to the minority class. In the second scenario, for K greater than or equal to 11, the K-NN algorithm assigns a positive value to the test vector, despite it being very close to the negative category vectors. In the third scenario, when the number of neighbors in the negative category is equal to the number of neighbors in the positive category, the K-NN algorithm assigns a random value to the test vector.

To address this issue and minimize the number of misclassified cases by the K-NN algorithm, the E-KNN algorithm revises the majority vote approach by adopting a sophisticated mechanism that adjusts distances based on the class distribution and the proximity of neighboring points. This section outlines the steps of the E-KNN algorithm and the improvements made over the K-NN algorithm, explaining how they contribute to enhanced classification accuracy.

Calculation of Euclidean distance D(X,Y)*:*

Calculate the Euclidean distance that separates each test point X and its neighbors Y, using the Eq. (1).

•Sort all calculated distances D(X,Y) in ascending order.

•Select the K smallest distances to determine the K nearest neighbors.

Calculation of the probability factor P:

P is defined as the ratio of the number of neighbors in a specific class (N_c) to the total number of neighbors (K). The *P* factor is designed to weight the contribution of each neighbor based on the density and prevalence of their respective class in the immediate vicinity. The probability factor *P* is calculated by Eq. (3):

$$P = \frac{N_c}{K} \tag{3}$$

Calculation and adjustment of the new distance $D_n(X,Y)$

To calculate $D_n(X,Y)$, which measures the separation between each point X and its neighbors Y of the majority class, apply the following Eq. (4):

$$D_n(X,Y) = D(X,Y) - \left((D(X,Y) * P) + d\right) \quad (4)$$

To calculate $D_n(X,Y)$, which measures the separation between each point X and its neighbors Y of the minority class, apply the following formula Eq. (5):

$$D_n(X,Y) = D(X,Y) - \left((D(X,Y) * P) - d\right) \quad (5)$$

The use of the d parameter (distance adjustment parameter) is crucial. By adding or subtracting d, the algorithm adjusts the measured distance between points to avoid unfairly favoring either the overrepresented (majority classes) or underrepresented (minority classes). This helps ensure that all classes are treated fairly in the classification process, thus improving the overall accuracy of the algorithm.

Sorting and classification

After calculating all the new distances $D_n(X,Y)$, sorting them in ascending order allows the algorithm to determine the most influential neighbor for classifying the test vector X, selecting the one with the lowest $D_n(X,Y)$. This method ensures that the class assigned to X is represented by not only the closest neighbor but also the most statistically relevant. This approach enhances the integrity and increases the overall accuracy of the model.

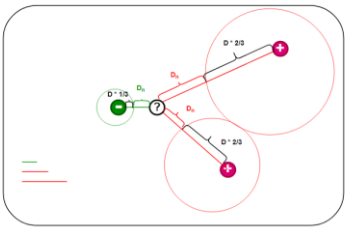
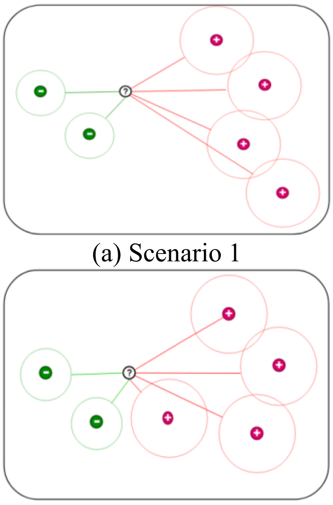


Figure 11. Calculation of new distance D_n



(a) Scenario 1

(b) Scenario 2

Figure 12. Class prediction by E-KNN

To illustrate the mechanism of weighted distance and the classification of a test point X. Figure 11 shows an illustrative example of the distance calculation $\boldsymbol{D_n}$. For this scenario, we have K=3 neighbors (two positive neighbors and one negative neighbor). The probability of assigning the positive class to the tested vector is $p=\frac{2}{3}$. To calculate the new distance $\boldsymbol{D_n}$ that separates the tested vector and each element of the positive class, we use the formula $\boldsymbol{D_n=D-((D*\frac{2}{3})+d)}$. The probability of assigning the negative class to the tested vector is $p=\frac{1}{3}$. To calculate the new distance $\boldsymbol{D_n}$ that separates the tested vector and each element of the negative class, we use the formula $\boldsymbol{D_n=D-((D*\frac{1}{3})-d)}$.

Figure 12 shows an illustrative example of class prediction by E-KNN algorithm.

In scenario (1), we have K=6 (4 positive neighbors and 2 negative neighbors). After calculating and sorting the new distances $\boldsymbol{D_n}$, the tested vector is assigned to the negative category, which is a minority category. In scenario (2), we have K=6 (4 positive neighbors and 2 negative neighbors). After calculating and sorting the new distances $\boldsymbol{D_n}$, the tested vector is assigned to the positive category, which is the majority category. The pseudo code of the E-KNN is as follows:

```
Input: Training dataset, test dataset, K, d
Output: Class of test vector X
Begin
For each test vector X in the Test dataset
For each training vector Y in the Training dataset
Calculate the distance between X and Y, denoted as D(X,Y), using Eq. (1).
End For
Sort all calculated distances D(X, Y) in increasing order.
Select the K smallest distances to determine the K closest neighbors.
Compute the probability P of assigning the neighbor Y's class to the test vector X based on the distribution of the K neighbors using Eq. (3).
For each of the K neighbors
Calculate the new distance Dn(X,Y) using formula Eq. (4) for the majority class and formula Eq.(5) for the minority class.
End For
Sort the new distances Dn(X,Y) in ascending order and select the smallest distance and the neighbor corresponding to that distance.
Assign the tested vector to the category of the selected neighbor.
End For
End
```

3.2.5 Performance evaluation

Evaluating performance is crucial for determining the precision and efficacy of various models, thereby facilitating the selection of the most appropriate model for the given task. To assess the effectiveness of the E-KNN classifier on the dataset in question, we compute key performance indicators including accuracy, recall, precision, and F-measure. Also, we used error metrics such as MAE and RMSE for performance evaluation. To represent the projected classifier efficiency, it is compared with five other classifiers, namely, KNN, LR, RF, DT and SVM. To calculate the performance evaluation metrics, we used the confusion matrix presented in Table 3.

Table 3. Confusion matrix

		Predicted	
		Negative (0)	Positive (1)
Actual	Negative (0)	TN	FP
	Positive (1)	FN	TP

True Positive (TP): refers to the count of instances accurately identified as positive.

False Positive (FP): denotes the count of instances erroneously labeled as positive.

True Negative (TN): signifies the count of instances correctly categorized as negative.

False Negative (FN): indicates the count of instances mistakenly categorized as negative.

Performance metrics can be measured from the values of TN, TP, FN and FP by the equations Eq. (6), Eq. (7), Eq. (8) and Eq. (9). The error metrics are calculated by the Eq. (10) and Eq. (11).

Accuracy assesses the fraction of accurate forecasts generated by the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Precision quantifies the ratio of correct positive predictions out of all positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Recall assesses the fraction of correct positive predictions out of all actual positives.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

F-measure is a combination of precision and recall that provides an overall measure of model performance.

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

MAE is the average of the absolute differences between the actual (y_i) and predicted values $(\hat{y}_l)$.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_l|}{n} \tag{10}$$

RMSE is computed by finding the square root of the average of the squared discrepancies between the predicted values $(\hat{y}_l)$ and the actual values (y_i)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_l - y_i)^2}{n}} \tag{11}$$

4. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed E-KNN classifier is compared to other algorithms cited previously based on the metrics discussed in subsection 3.2.5. Table 4 and Figures 13-16 show the performance results of the comparison of different classifiers.

The introduction of weighted distance calculations and class distribution sensitivity in E-KNN has a direct and profound impact on its performance metrics. E-KNN achieves a high accuracy of 91.43%, significantly better than standard K-NN. This improvement is largely due to the weighted distance calculation which ensures that predictions are more influenced by neighbors that are closer and potentially more relevant, thereby reducing classification errors. The precision of E-KNN at 94.12% and recall at 88.89% are indicative of the model's ability to correctly identify positive cases while minimizing false positives. The class distribution sensitivity plays a crucial role here, ensuring that minority classes are adequately represented in the decision-making process, thus improving the detection of true positives and reducing false negatives. The balanced F-measure of 91.43% reflects the model’s effectiveness in maintaining a harmonious balance between precision and recall, a crucial aspect in medical applications where both avoiding false negatives and minimizing false positives are important.

Table 4. Evaluation of performance metrics

Classifier/Metrics	Accuracy (%)	Recall (%)	Precision (%)	F-Measure (%)
SVM	88.57	91.67	86.84	89.19
LR	87.14	88.89	86.49	87.67
DT	84.28	83.34	85.71	84.51
RF	85.71	88.89	84.21	86.49
K-NN	88.57	88.89	88.89	88.89
E-KNN	**91.43**	**88.89**	**94.12**	**91.43**

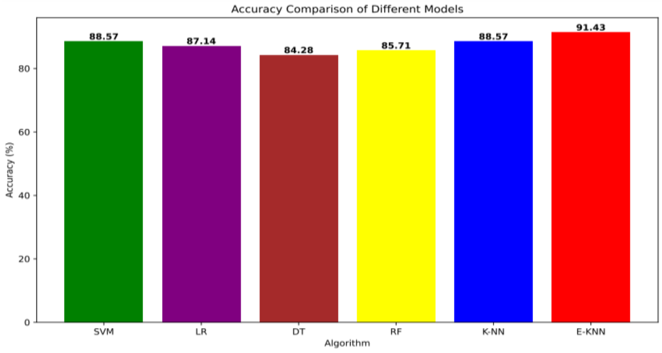


Figure 13. Accuracy comparison

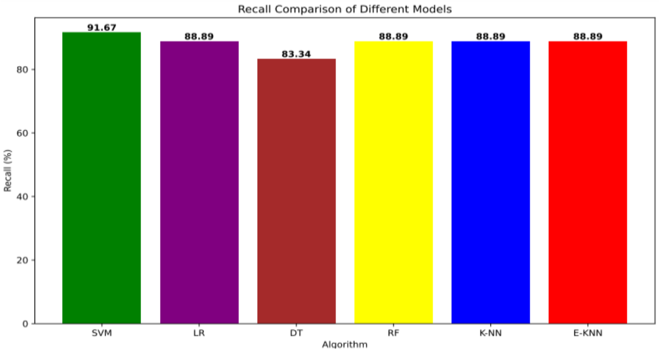


Figure 14. Recall comparison

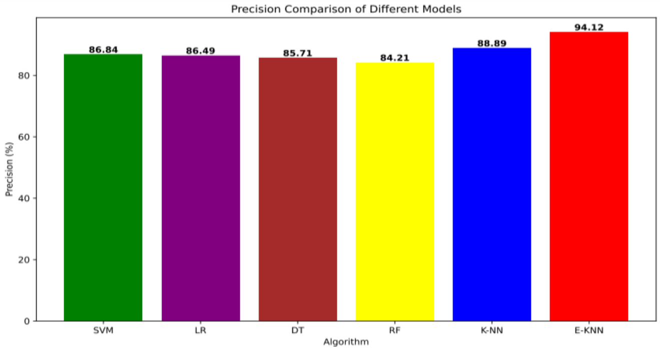


Figure 15. Precision comparison

Table 5. Evaluation of error metrics

Classifier/Metrics	MAE	RMSE
SVM	0.11	0.34
LR	0.13	0.38
DT	0.16	0.40
RF	0.14	0.38
K-NN	0.11	0.34
E-KNN	**0.08**	**0.29**

Table 5 and Figures 17, 18 show the error metrics results comparison of E-KNN vs. different classifiers. We can see that the E-KNN MAE and RMSE are lower than the error metrics of other classifiers, which were 0.08, 0.29, respectively. These lower error rates prove the effectiveness of the weighted distance calculation in enhancing the overall predictive accuracy and consistency of the model, even in the presence of outliers or anomalous data. We conclude that the E-KNN has a higher capacity for reducing disparities between predictions and observations.

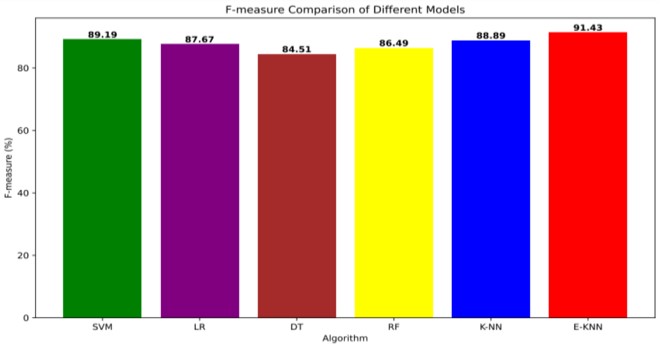


Figure 16. F-measure comparison

The technical enhancements of E-KNN not only address the limitations of traditional K-NN but also significantly enhance its utility in clinical settings. The improvements in accuracy, precision, recall, and error metrics validate the effectiveness of the weighted distance calculation and class distribution sensitivity, highlighting E-KNN's superior capability to deliver reliable and accurate predictions in the early detection and treatment of CVD.

The proposed algorithm for CVD prediction, E-KNN, is compared with various studies recently proposed by researchers to contribute to diagnosing a CVD with precision, perfection, and efficiency such as [21, 22, 26, 27, 32, 33]. The results are given in Table 6 and Figure 19.

The E-KNN ML model shows commendable performance when compared to models referenced in [21, 22, 26, 27, 32] across several critical metrics. Notably, E-KNN's accuracy of 91.43% is superior to all the aforementioned models except for the CNN model [33], which slightly edges out E-KNN with an accuracy of 92.89%. Additionally, E-KNN's recall rate of 88.89% is surpassed only by CNN's recall of 94%, indicating significant improvements over the other compared models.

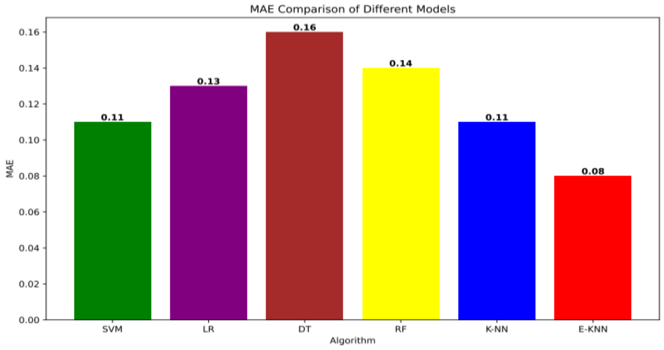


Figure 17. MAE error comparison

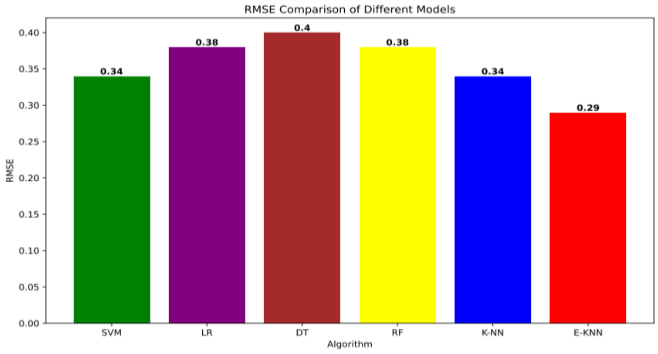


Figure 18. RMSE error comparison

Table 6. Performance comparison: E-KNN vs. prior research

Research Work	ML Model	Accuracy (%)	Recall (%)	Precision (%)	F-Measure (%)
[21]	RF	90.16	88.20	93.7	90.9
[22]	SVM	89.34	80.70	95.83	87.61
[26]	SMO	86.47	86.5	86.5	86.4
[27]	Hybrid model DT+RF	88	-	-	-
[32]	RF	88.10	78.95	93.75	85.71
[33]	CNN	92.89	94	94	94
Proposed work	E-KNN	**91.43**	**88.89**	**94.12**	**91.43**

In terms of precision, E-KNN records a precision of 94.12%, slightly higher than the CNN's precision of 94%. This slight edge demonstrates E-KNN's effective minimization of false positives, crucial in medical applications where precise diagnostics are required. However, when examining the F-measure, which combines both precision and recall to provide a balanced view of model performance, E-KNN's F-measure stands at 91.43%, slightly below the CNN's 94%. This difference points to an area where E-KNN, despite its high precision, could improve in balancing detection of true positives while minimizing false positives as effectively as the CNN model.

Despite E-KNN's slightly lower F-measure, its overall performance remains impressive, particularly in terms of precision and its capability in the robust prediction of CVD. The E-KNN model thus significantly enhances the overall accuracy and reliability of the CVD prediction system, asserting its utility as a potent tool for clinical applications.

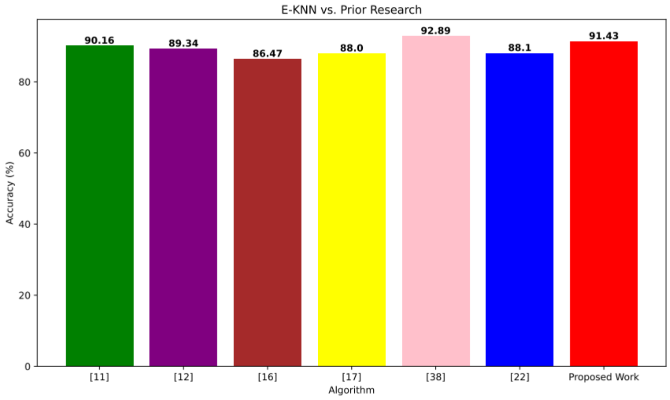


Figure 19. Comparing accuracy: E-KNN vs. prior research

5. CONCLUSION AND FUTURE WORK

Cardiovascular diseases are a global healthcare challenge, and their accurate diagnosis is crucial. Any slight error in diagnosis can have severe consequences. Given the rising mortality rate from these diseases, there is a need to create an intelligent cardiac disease prediction system. This system would monitor patients' real-time physiological signals using wearable sensors.

This study introduces a novel CVD prediction model based on an enhanced version of the K-NN algorithm, termed E-KNN. The E-KNN algorithm builds upon the standard K-NN, a non-parametric classification method based on the majority vote of an observation's nearest neighbors. E-KNN addresses the shortcomings of K-NN in scenarios with unbalanced medical datasets, where minority class data may skew predictions. It enhances accuracy by incorporating a probability factor (P) that weights the contribution of each neighbor based on class density and prevalence. Additionally, it adjusts distances with a parameter (d) to ensure fair treatment of both majority and minority classes, refining the traditional majority voting approach. The effectiveness of the E-KNN model is evaluated and compared with several established models using K-NN, RF, DT, LR, and SVM.

Another objective of this study is to establish an IoT platform equipped with biomedical sensors for gathering crucial medical data, including an Arduino board, a blood pressure sensor, a pulse sensor, and an ECG sensor. The data collected from these sensors are processed by the E-KNN-based model to produce more precise predictions for CVD.

The E-KNN algorithm achieved the highest accuracy of 91.43%, outperforming SVM (88.57 %), LR (87.14 %), DT (84.28 %), RF (85.71 %), and KNN (88.57%). This indicates that E-KNN is more successful in correctly classifying patients with and without CVD. Examining the error metrics, E-KNN exhibited the lowest Mean Absolute Error of 0.08 and Root Mean Squared Error of 0.29, highlighting its superior performance in terms of prediction errors.

In the future, the work could be improved by developing a mobile application based on the E-KNN algorithm. Testing the E-KNN on a larger dataset could enhance the accuracy and reliability of the results. Additionally, employing feature selection techniques to choose the most relevant attributes could boost the performance of the E-KNN for CVD prediction. Future research could also explore the use of E-KNN in various fields, such as diabetes prediction, cancer detection, financial fraud identification, and consumer behavior analysis, providing valuable insights into the comparative effectiveness of E-KNN.

REFERENCES

[1] Chui, K.T., Alhalabi, W., Pang, S.S.H., Pablos, P.O.D., Liu, R.W., Zhao, M. (2017). Disease diagnosis in smart healthcare: Innovation, technologies and applications. Sustainability, 9(12): 2309. https://doi/org/10.3390/su9122309

[2] Moshawrab, M., Adda, M., Bouzouane, A., Ibrahim, H., Raad, A. (2023). Smart wearables for the detection of cardiovascular diseases: A systematic literature review. Sensors, 23(2): 828. https://doi.org/10.3390/s23020828

[3] APS: Algeria Press Services webpage. https://www.aps.dz/en/health-science-technology/.

[4] Sonune, S., Kalbande, D., Yeole, A., Oak, S. (2017). Issues in IoT healthcare platforms: A critical study and review. In 2017 International Conference on Intelligent Computing and Control (I2C2), Coimbatore, India, pp. 1-5. https://doi.org/10.1109/I2C2.2017.8321898

[5] Oracle's Website. https://www.oracle.com/dz/internet-of-things/what-is-iot/.

[6] Reddy K, S., Sidaarth R., Reddy, S.A., Shettar, R. (2019). IoT based health monitoring system using machine learning. IJARIIE-ISSN(O)-2395-4396, 5(3): 381-386. https://ijariie.com/AdminUploadPdf/IoT_based_Health_Monitoring_System_using_Machine_Learning_ijariie10244.pdf.

[7] Islam, M.N., Raiyan, K.R., Mitra, S., Mannan, M.R., Tasnim, T., Putul, A.O., Mandol, A. B. (2023). Predictis: an IoT and machine learning-based system to predict risk level of cardio-vascular diseases. BMC Health Services Research, 23(1): 171. https://doi.org/10.1186/s12913-023-09104-4

[8] Bhardwaj, R., Nambiar, A.R., Dutta, D. (2017). A study of machine learning in healthcare. In 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, Italy, pp. 236-241. https://doi.org/10.1109/COMPSAC.2017.164

[9] Samuel, A.L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 3(3): 210-229. https://doi.org/10.1147/rd.33.0210

[10] Kalaiselvi, K., Deepika, M. (2020). Machine learning for healthcare diagnostics. In Machine Learning with Health Care Perspective: Machine Learning and Healthcare, pp. 91-105. https://doi.org/10.1007/978-3-030-40850-3_5

[11] Panesar, A. (2022). Machine Learning and AI for Healthcare: Big Data for Improved Health Outcomes, Second Edition, Apress: London, UK.

[12] Wang, L. (2005). Support Vector Machines: Theory and

Applications, Springer Science & Business Media, Vol. 177.

[13] Shi, L., Duan, Q., Ma, X., Weng, M. (2012). The research of support vector machine in agricultural data classification. In Computer and Computing Technologies in Agriculture V: 5th IFIP TC 5/SIG 5.1 Conference, CCTA 2011, Beijing, China, pp. 265-269. https://doi.org/10.1007/978-3-642-27275-2_29

[14] Podgorelec, V., Kokol, P., Stiglic, B., Rozman, I. (2002). Decision trees: An overview and their use in medicine. Journal of Medical Systems, 26: 445-463. https://doi.org/10.1023/A:1016409317640

[15] Barros, R.C., Basgalupp, M.P., Freitas, A.A., De Carvalho, A.C. (2013). Evolutionary design of decision-tree algorithms tailored to microarray gene expression data sets. IEEE Transactions on Evolutionary Computation, 18(6): 873-892. https://doi.org/10.1109/TEVC.2013.2291813

[16] Biau, G. (2012). Analysis of a random forests model. The Journal of Machine Learning Research, 13(1): 1063-1095

[17] Tripoliti, E.E., Fotiadis, D.I., Manis, G. (2011). Automated diagnosis of diseases based on classification: Dynamic determination of the number of trees in random forests algorithm. IEEE Transactions on Information Technology in Biomedicine, 16(4): 615-622. https://doi.org/10.1109/TITB.2011.2175938

[18] Bisong, E., & Bisong, E. (2019). Logistic regression. In Building Machine Learning and Deep Learning Models on Google Cloud Platform, pp. 243-250. https://doi.org/10.1007/978-1-4842-4470-8_20

[19] Chomboon, K., Chujai, P., Teerarassamee, P., Kerdprasop, K., Kerdprasop, N. (2015). An empirical study of distance metrics for k-nearest neighbor algorithm. In Proceedings of the 3rd International Conference on Industrial Application Engineering, Kitakyushu, Japan, pp. 280-285. https://doi.org/10.12792/iciae2015.051

[20] Cover, T., Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1): 21-27. https://doi.org/10.1109/TIT.1967.1053964

[21] Rajdhan, A., Sai, M., Agarwal, A., Ravi, D., Ghuli, P. (2020). Heart disease prediction using machine learning. International Journal of Engineering Research & Technology, 9(4): 659-662

[22] Ware, S., Rakesh, K.R., Choudhary, B. (2020). Heart attack prediction by using machine learning techniques. International Journal of Recent Technology and Engineering, 8(5): 1577-1580. https://doi.org/10.35940/ijrte.D9439.018520

[23] Magar, R., Memane, R., Raut, S. (2020). Heart disease prediction using machine learning. Journal of Emerging Technologies and Innovative Research, 7(6): 2081-2085. http://www.jetir.org/papers/JETIR2006301.pdf.

[24] Shah, D., Patel, S., Bharti, S.K. (2020). Heart disease prediction using machine learning techniques. SN Computer Science, 1(6): 345. https://doi.org/10.1007/s42979-020-00365-y

[25] Arghandabi, H., Shams, P. (2020). A Comparative study of machine learning algorithms for the prediction of heart disease. International Journal for Research in Applied Science & Engineering Technology, 8(12): 677-683. http://doi.org/10.22214/ijraset.2020.32591

[26] Reddy, K.V.V., Elamvazuthi, I., Aziz, A.A., Paramasivam, S., Chua, H.N., Pranavanand, S. (2021). Heart disease risk prediction using machine learning classifiers with attribute evaluators. Applied Sciences, 11(18): 8352. https://doi.org/10.3390/app11188352

[27] Kavitha, M., Gnaneswar, G., Dinesh, R., Sai, Y.R., Suraj, R.S. (2021). Heart disease prediction using hybrid machine learning model. In 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, pp. 1329-1333. https://doi.org/10.1109/ICICT50816.2021.9358597

[28] Chowdhury, M.N. R., Ahmed, E., Siddik, M.A.D., Zaman, A.U. (2021). Heart disease prognosis using machine learning classification techniques. In 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, pp. 1-6. https://doi.org/10.1109/I2CT51068.2021.9418181

[29] Manaa, A., Brahimi, F., Chouiref, Z., Kessouri, M., Amad, M. (2022). Cardiovascular diseases prediction based on dense-DNN and feature selection techniques. In International Symposium on Modelling and Implementation of Complex Systems, Mostaganem, Algeria, pp. 333-347. https://doi.org/10.1007/978-3-031-18516-8_24

[30] Pan, C., Poddar, A., Mukherjee, R., Ray, A.K. (2022). Impact of categorical and numerical features in ensemble machine learning frameworks for heart disease prediction. Biomedical Signal Processing and Control, 76: 103666. https://doi.org/10.1016/j.bspc.2022.103666

[31] Almulihi, A., Saleh, H., Hussien, A.M., et al. (2022). Ensemble learning based on hybrid deep learning model for heart disease early prediction. Diagnostics, 12(12), 3215. https:/doi/org/10.3390/ diagnostics12123215

[32] Jansi Rani, S.V., Chandran, K.S., Ranganathan, A., Chandrasekharan, M., Janani, B., Deepsheka, G. (2022). Smart wearable model for predicting heart disease using machine learning: Wearable to predict heart risk. Journal of Ambient Intelligence and Humanized Computing, 13(9): 4321-4332. https://doi.org/10.1007/s12652-022-03823-y

[33] Umer, M., Sadiq, S., Karamti, H., Karamti, W., Majeed, R., Nappi, M. (2022). IoT based smart monitoring of patients' with acute heart failure. Sensors, 22(7): 2431. https://doi.org/10.3390/s22072431

[34] Subahi, A.F., Khalaf, O.I., Alotaibi, Y., Natarajan, R., Mahadev, N., Ramesh, T. (2022). Modified self-adaptive Bayesian algorithm for smart heart disease prediction in IoT system. Sustainability, 14(21): 14208. https://doi.org/10.3390/su142114208

[35] Djerioui, M., Brik, Y., Ladjal, M., Attallah, B. (2019). Neighborhood component analysis and support vector machines for heart disease prediction. Ingénierie des Systèmes d'Information, 24(6): 591-595. https://doi.org/10.18280/isi.240605

[36] IEEE Xplore Webpage. https://ieeexplore.ieee.org/document/4122029.

[37] Springer Link Webpage. https://link.springer.com/article/10.3758/s13428-020-01516-y.

[38] Heart Disease Dataset. https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

[39] Wang, Z., Yao, L., Li, D., Ruan, T., Liu, M., Gao, J. (2018). Mortality prediction system for heart failure with orthogonal relief and dynamic radius means.

International Journal of Medical Informatics, 115: 10-17. https://doi.org/10.1016/j.ijmedinf.2018.04.003

[40] StandardScaler Technique. https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html/.