Medical Data Analytics Model: Developing A Multi-Model Framework for Preprocessing Multi-Modality Medical Datasets for Enhancing Quality

[1]Suresh Kapare, [2]Dr. V. Maria Anu
[1] Research Scholar, Sathyabama Institute of Science and Technology, Chennai.
[2] Associate Professor, VIT, Chennai.
Email-ID: [1]kaparesuresh8@gmail.com, [1]sureshkapare2012@gmail.com , [2]mariaanu.v@vit.ac.in

Abstract
This work presents a novel learning model framework for preprocessing medical data to improve data quality. The medical industry generates a diverse range of data, including time series, images, signals, and videos. However, the data from medical instruments, devices, sensors, and video-image cameras is often noisy due to environmental factors and patient-device distance. The metadata for patient, healthcare, and medical data is continuously updated to reflect disease progression and the patient's living conditions. Additionally, patients must undergo various tests and diagnostic procedures over an extended period to determine the disease stage and receive appropriate treatment. It creates more data quality problems, such as data redundancy, missing elements, low resolution, and noisy data. That can be rectified to provide an accurate prediction in medical data diagnosis. This paper offers a unique learning model framework for preprocessing patient data. It focused on EEG, ECG, EMG, MRI, and regularized body mass index data, such as heart rate and oxygen saturation, which were used to experiment with the framework. The framework employs multiple learning algorithms to preprocess various types of data. The experimental results, a testament to the framework's effectiveness, demonstrate that it can provide 99.89% quality data ready for manipulation. The results show that the proposed framework outperforms the others.
Keywords: Medical Data, Preprocessing, Machine Learning Algorithms, Image Preprocessing, Signal Preprocessing, Time-Series Preprocessing, EEG, ECG, MRI.
How to cite this article: Kapare S, Maria Anu V. Medical Data Analytics Model: Developing A Multi-Model Framework for Preprocessing Multi-Modality Medical Datasets for Enhancing Quality. Int J Drug Deliv Technol. 2026;16(66s):272-285. DOI: 10.25258/ijddt.16.66s.34

1. Introduction

Improving the accuracy and quality of medical data is a complex task, particularly given the data's volume, noise, and critical nature [1]. Machine learning (ML) approaches have emerged as essential technologies for preprocessing medical data. However, the data's complex, imperfect, and multidimensional nature presents significant challenges in accurately detecting diseases and other issues at earlier stages. In this context, data preprocessing in ML plays a crucial role. Its main objective is to deliver data processed with enhanced accuracy, thereby improving the precision of machine learning models. Data preprocessing involves transforming or encoding data into a format that machines can understand. The primary goal of developing an algorithm is to generate reliable, precise predictions, enabling it to analyze the data's characteristics [2]. Due to their various sources, many real-time ML datasets may be missing, inconsistent, or inaccurate. Data mining techniques are often inadequate as they cannot efficiently detect patterns. For this reason, data preprocessing is essential for enhancing the overall quality of the data. Missing or duplicate values can create a misleading view of the data's overall statistics. Figure-1 illustrates the primary steps in data preprocessing: cleaning, integration, transformation, and reduction.

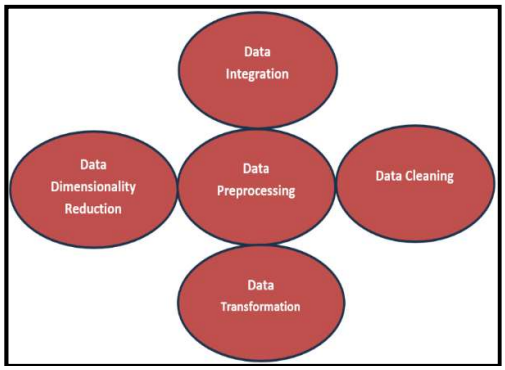


Figure-1. Common Pre-Processing Steps

Data Cleaning is carried out as part of pre-processing, where outliers are removed, inconsistencies are resolved, numerous noisy data points are reduced, and missing values are filled [3]. In measuring a variable, reducing noisy data involves removing uncertainty or random error using techniques such as binning, regression, and clustering. Data integration is essential while solving real-world issues, such as identifying nodules in a CT scan image [4]. Integrating images from various medical nodes is the only possible

approach to creating a larger database. Challenges may arise when applying data integration as a pre-processing step. Multiple forms with different features among the data cause difficulties in integration. Eliminating unwanted features across all data sources and identifying and resolving inconsistencies in data values are additional challenges [5]. After cleaning, the quality data is combined into other forms by modifying its characteristics through the data transformation step [6]. There are several strategies involved in transformation, including generalization, normalization, attribute selection, and aggregation. The generalization strategy utilizes concept hierarchies to convert the low-level data into high-level data [7]. On the other hand, the numerical features are adjusted to fit into a given range. This approach limits data attributes to a specific case to establish a correlation across various data points. Min-max, Z-score, and decimal are a few normalization methods [8].

According to algorithms, handling a vast data set is challenging. It is essential to obtain simplified data sets with minimal quantities that provide ideal output quality to meet such challenges. Data reduction techniques offer solutions for this issue. They encompass various strategies, such as data cube aggregation, dimension reduction, data compression, discretization, numerosity reduction, and attribute subset selection. [9] Feature extraction is performed using dimensionality reduction techniques. The characteristics of the data are referred to as the dataset's dimensionality. The primary objective of this method is to eliminate redundant features that machine learning algorithms consider. The implementation of encoding technology enables better reductions in data size. Conversely, data compression can be lossy or lossless. Lossless compression allows the recovery of the original data, which is not possible with lossy compression. Data discretization categorizes continuous attributes into discrete intervals. This process is essential because continuous features are less likely to correlate with the target variable. After variable discretization, analyzing the groups related to the target becomes feasible, although interpreting the results can be quite challenging. In numerosity reduction, the data representation can be an equation or a model, such as a regression model [10]. This approach reduces the effort required to store a model rather than massive amounts of data. Attribute selection is also critical. Improper selection leads to high-dimensional data, causing training challenges due to fitting issues. Only the attributes that add value for training are considered.

The following contributions enhance the overall efficacy of the medical data preprocessing output.

- A deep discussion is presented to make the reader understand various processes of preprocessing medical data.
- The major classes and their internal functions of data preprocessing are explained in detail and shown in a pictorial representation.
- Data preprocessing is demonstrated with different forms of medical data.

2. Literature Survey

In this section, various previous research efforts on data preprocessing methods proposed by multiple authors for the medical sector are discussed in detail. Traditionally, many experts highly recommend that ML-based algorithms process various types of input datasets. Recently, deep learning-based algorithms have gained widespread use in processing raw input data. For example, Kanani & Padole (2020) stated that Electrocardiogram (ECG) signals are the most efficient means of monitoring the cardiovascular system's performance and health. Automated categorization and detection of ECG arrhythmia signals, which yield faster and more reliable outcomes, have become essential. Various machine learning techniques enhance accuracy while ensuring high model speed and resilience. This study proposes a preprocessing method that significantly improves the accuracy of deep learning models used for ECG classification, combined with a modified deep learning architecture that enhances training performance. This preprocessing approach, combined with the system's deep learning model, maintains accuracy exceeding 99% without overfitting.

Nancy, A., et al. (2022) have researched the use of artificial intelligence and machine learning technologies for statistical analysis in the healthcare industry. Accurate disease prediction is crucial for implementing preventive measures and early detection in individuals at risk. Bi-LSTM (bidirectional long short-term memory) underpins an innovative healthcare system that monitors and accurately predicts the risk of heart disease. It has impressive success metrics, outperforming existing innovative heart disease prediction platforms by 98.86%, 98.9%, 98.8%, 98.89%, and 98.86% F-measure. Bandyopadhyay et al. (2021) have proposed a brief overview of deep learning and machine learning processes. Three popular algorithms are utilized in both techniques: Bagged Tree Ensemble, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM). Identifying the type of skin disease in the medical field is challenging. Sometimes, determining the exact type of disease can be difficult due to the complexity of human skin texture and the visual effects of the conditions. Artificial intelligence (AI) is

experiencing significant expansion in the healthcare industry today. These techniques streamline the process and improve testing methods.

Mishra S et al. (2021) presented a comprehensive study focusing on computer-aided diagnosis and detection, a dynamic research topic that continues to grow in the significant field of medical diagnosis and detection. AI algorithms can identify and diagnose various diseases, including lung disease, rheumatoid arthritis, heart disease, Alzheimer's disease, hepatitis, dengue, and Parkinson's disease. Large neural networks, consisting of interconnected neurons, can adjust their hyperparameters based on newly updated data in deep learning. This study evaluates the effectiveness of deep learning in identifying specific disease risk factors. Additionally, two distinct case studies are reviewed in detail to illustrate the role of deep learning methods in disease diagnosis. Roy, T. S., et al. (2022) propose CNN-based algorithms for the identification of valvular heart diseases. It presents the first-ever CNN-based Xception model for valvular heartbeat analysis. This model achieved a 99.45% accuracy rate on the experimental dataset, with 98.5% sensitivity and 98.7% specificity. In the proposed modified Xception network model, the testing duration is the shortest, and the accuracy in predicting the occurrence of valvular heart disease is the highest among deep learning models, including LeNet-5, AlexNet, VGG16, and VGG19. SVM and Random Forest algorithms ranked highest among machine learning techniques across all deep learning approaches, indicating they can be adapted to the CNN-based Xception model for optimal performance.

Usman, S. M., et al. (2020) propose using deep learning techniques to develop a seizure prediction system by preprocessing scalp EEG signals, automatically extracting features with convolutional neural networks, and finally categorizing the data with support vector machines. The proposed strategy achieved average sensitivities and specificities of 92.7% and 90.8%, respectively, across 24 subjects from the CHBMIT scalp EEG dataset. Gawhale et al. (2023) presented a review of an innovative approach to detecting sliding disorder using a pair of machine-learning classifiers and a hybrid deep-learning framework. The first step involves collecting EEG signals from patients with sleep disorders and healthy controls. The proposed methodology is designed for implementation on embedded devices to provide a novel and efficient method for classifying sleep stages. The research utilizes the Power Spectrum Density (PSD) dataset for testing. The results indicate that the Ensemble Machine Learning Classifier paired with Hybrid Deep Learning significantly improves classification accuracy to 96.78% and sensitivity to 89.06%.

Alvi, A. M., et al. (2021) conducted research to develop an EEG data analysis system that uses a multi-layer Gated Recurrent Unit (GRU) for anomaly detection, addressing these issues. The proposed framework consists of four steps: (1) gathering raw EEG data; (2) pre-processing the data (3) applying a GRU-based method to classify and uncover significant properties of EEG data; and (4) evaluating the model's performance using a publicly accessible EEG dataset, where it achieved 96.91% accuracy, 97.95% sensitivity, 96.16% specificity, and 96.39% F1 score. Li J. P. et al. (2020) propose a heart disease detection model that uses various classification algorithms, including SVM, LR, ANN, KNN, NB, and DT. To tackle the feature selection problem, a unique Fast Conditional Mutual Information approach is employed to enhance classification accuracy and reduce the time required for the classification system to execute. The classifiers' performance is evaluated based on the results of the reading actions and the features selected by the algorithms. Based on the experimental results, it is possible to create a high-level intelligent system for diagnosing heart disease using the proposed Feature Selection Method (FCMIM) in conjunction with a Support Vector Machine classifier. By comparing the proposed diagnostic methodology (FCMIM-SVM) to other methods that have demonstrated excellent accuracy, it is also possible to create a high-level intelligent system for diagnosing heart disease. Krori Dutta et al. (2023) proposed a machine learning approach for epilepsy detection. This work categorizes epileptic episodes into various classes using several machine learning and deep learning techniques, with time- and frequency-domain preprocessing. Temple University Hospital EEG (TUHEEG) was used to determine the ictal period of different seizures. The pre-processed data was then evaluated using a variety of classifiers, including SVC, KNN, and LSTM. The ictal phase EEG data of epileptic seizures have been processed using both time-domain and frequency-domain techniques and subsequently classified using a range of ML and DL methods. The KNN achieved the best results.

2.1 Limitations and Motivations

The observations from the survey above indicate a need for an advanced methodology to diagnose diseases using various input data samples. Achieving early diagnosis of diseases with high accuracy remains a significant challenge in prior research. Another critical issue faced by the medical industry is the complexity of data when using traditional methods. Although the existing model discussed in the previous section has achieved high accuracy, it struggles to process large datasets. This paper explores recent data pre-processing steps to address these challenges. Data pre-processing is one of the most effective techniques for minimizing and eliminating artifacts from input samples. This will

enhance the accuracy rate of the proposed model for disease classification.

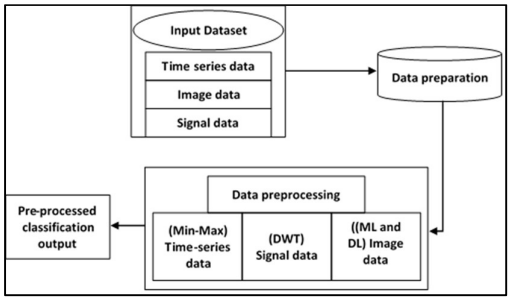


Figure-2: Proposed Workflow

3. Proposed Methodology

This paper examines various types of medical data, including time-series, image, and signal-based data, to accurately detect both normal and abnormal samples through multiple preprocessing steps. The pre-processing methods also differ according to the type of input data samples. The overall workflow of the proposed approach consists of the following steps: data preparation, data pre-processing, and classification result. The overall workflow of the proposed model is illustrated in Figure-2. The following section provides an in-depth explanation of the function of each data-processing step employed in this paper.

3.1 Data preparation

Data preparation involves converting raw input data into a suitable format to facilitate model learning from the input samples. The overall input data is divided into two phases: training and testing. For training, 80% of the data is used, while the remaining 20% is used for testing. Based on the results of the trained and tested data, the final classification result of the proposed approach is determined.

3.2 Data Preprocessing in Data Mining

Data preprocessing is essential for organizations that collect raw data and transform it into various formats for specific tasks. The primary goal of data preprocessing is to enhance data quality and make it more suitable for specific data mining tasks. This step-by-step procedure includes collecting, filtering, sorting, processing, storing, and presenting data in a readable format. Typically, this work is carried out by data scientists, specialists, or data engineers within any organization. The common steps in data preprocessing, along with their internal functions, are explained below and illustrated in Figure 3.

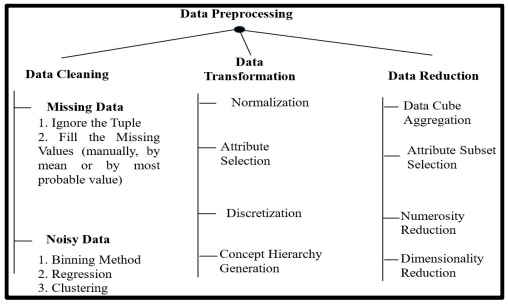


Figure-3. Architecture Of Preprocessing Steps

3.3 KNN algorithm

One of the simplest machine learning algorithms is the K-Nearest Neighbors (KNN) algorithm, which is a supervised learning method. The K-NN method classifies a new case based on the group nearest to it among the existing categories. The latest instance and its data resemble already present examples. The K-NN method categorizes a new data point based on its similarity to existing data points, collecting all relevant data. This indicates that the K-NN algorithm can assign new data to an appropriate category. The K-NN technique is primarily used for classification tasks and regression. It is also known as a lazy learner algorithm because, instead of learning immediately from the training set, it retains the dataset and processes it later to identify patterns. The K-NN method stores the dataset during training and classifies new data into the group that best matches the stored data. The K-NN algorithm operates through several steps outlined below. The first step is to select the number of K neighbors. The second step involves calculating the K number of neighbors based on Euclidean distance. The KNN algorithm is applied in the third step, using the Euclidean distance. The fourth step counts the number of data points in each category using K neighbors. In the fifth step, new data points are assigned to the category with the most neighbors. Finally, the model is ready for implementation. The KNN algorithm offers several advantages, including its ease of implementation, effective use of training data, and improved performance with larger datasets. Figure-5 shows the structure of the KNN Algorithm.

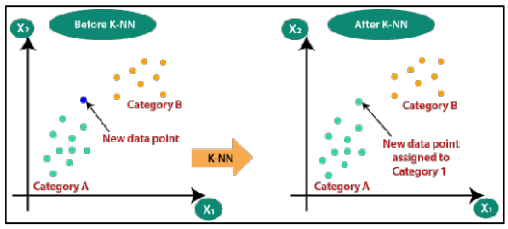


Figure 5: KNN Algorithm

3.4 Gradient Boosting

Gradient Boosting is a well-known machine learning algorithm for classification and regression tasks. Boosting is a type of ensemble learning algorithm that trains models sequentially, with each new model attempting to correct the errors of the previous one. It combines various weak learners into a single strong learner. Boosting algorithms are classified into two types: AdaBoost and Gradient Boosting. At each step of the process, the algorithm computes the gradient of the loss function with respect to the current ensemble's predictions, then trains a new weak model to minimize it. Next, AdaBoost, short for adaptive boosting, is an ensemble ML technique applicable to a wide range of regression and classification tasks. Figure-9 illustrates the structure of the Gradient Boosting algorithm.

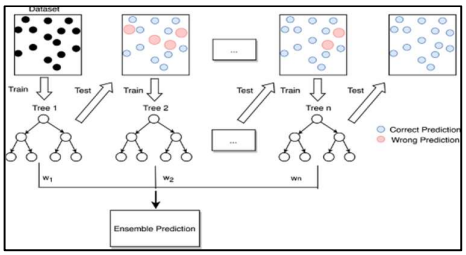


Figure-9. Gradient Boosting Algorithm

3.5 CNN algorithm

Data Preprocessing Using ML Algorithms

The explained ML models are implemented and evaluated on a multimodal dataset used in the medical industry.

KNN-based Medical Data Pre-Processing

For efficient data preprocessing, a custom K-Nearest Neighbors (KNN) model has been developed that encompasses several processes, including cleaning, integration, transformation, reduction, discretization, and normalization. The custom KNN method extends the conventional usage of KNN for classification and regression to act as a data-driven preprocessing tool. The KNN plays an essential role in each data preprocessing step, which is briefly explained with a mathematical model below:

1. Data Cleaning Using Custom KNN

In the given dataset, impute the noisy or missing values using the custom KNN. To fill in the missing records in the data point x_i by assuming the values its KNN in future space is the basic strategy. Let the dataset be denoted as $D = \{x_1, x_2, x_3 \dots, x_n\}$, where $x_i = x_{i1}, x_{i2}, \dots, x_{im}$ and few x_{ij} data points are missing. The missing values are calculated by:

$$x_{ij} = \frac{1}{K} \sum_{x_l \in N_k(x_i)} x_{lj}$$

Where the set of k-nearest neighbors of x_i is denoted as $N_k(x_i)$, the value of feature j in the neighbor x_l is denoted as x_{lj} and the mode is used for data when it is categorized. This strategy eliminates noise and fills in missing values by using similar patterns in the data.

2. Data Integration Using KNN-Based Feature Alignment

KNN is used to align features based on data type across datasets gathered from various sources, such as sensors, hospitals, and databases. Let's consider two datasets $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_m)$ which contains some overlapping data with slight contradictions. The custom KNN aligns $a_i \in A$ to $b_j \in B$ based on the nearest Euclidean distance:

$$match(a_i) = arg \min_{b_j \in B} \| a_i - b_j \|$$

Once the custom KNN aligns the dataset, the missing values in either dataset are filled with suitable data.

3. Data Transformation Using KNN Smoothing

Custom KNN reduces the fluctuations and noise for future transformation. The KNN smoothing is applied for instances where each value is replaced by the average value based on the neighbors.

$$\tilde{x}_{ij} = \frac{1}{k} \sum_{x_l \in N_k(x_i)} x_{lj}$$

The local smoothing function is performed by the $\tilde{x}_{ij}$ and also, it retains the underlying data manifold.

4. Data Reduction Using KNN Prototypes

To retain the decision boundaries while reducing the size of the dataset by using KNN-based reduction techniques like Condensed Nearest Neighbor (CNN) or Edited Nearest Neighbor (ENN).

The Condensed Nearest Neighbor (CNN) works by retaining the subset $S \subset D$, the KNN algorithm precisely classifies the subset S in the dataset D. Through this process, misclassified points are repeatedly added to S.

$$S_{t+1} = S_t \cup \{x_i \in D \mid KNN_S(x_i) = y_i\}$$

In the large volume of the dataset, this S_{t+1} can reduce the storage and computational time.

5. Discretization Using KNN Clustering

KNN helps with discretization by grouping continuous features into bins using neighborhood alignment, rather than predefined fixed intervals. For the (x_i) continuous feature requires developing a KNN graph, and density-based clustering needs to be applied, such as DBSCAN or hierarchical KNN, with bins assigned to clusters.

$Bin(x_i) = c$, if $x_{ij} \in Cluster_c$

This approach follows the data topology and performs better than equal-width or frequency binning.

6. Normalization Using KNN-Based Local Scaling

The KNN-based local scaling is used because it is more flexible and a better normalization than global min-max normalization for neighborhood statistics. In KNN, the feature x_{ij} is defined as: $\mu_k(x_{ij})$ is the mean of the j^{th} feature and $\sigma_k(x_{ij})$ is the standard deviation. After that, the Z-score normalization is applied,

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_k(x_{ij})}{\sigma_k(x_{ij})}$$

$\hat{x}_{ij}$ assures a better capture of local distribution and normalization.

GB-based ECG signal Data pre-processing (MAM)

The KNN algorithm preprocesses time-series data from the medical industry after customizing its internal functionality. However, it is unable to preprocess the signal or image dataset, even after customization. Hence, a Gradient Boosting algorithm is customized to preprocess ECG data, thereby enhancing signal quality and facilitating efficient disease prediction. GB models are effectively used for ECG signal preprocessing to enhance data quality by examining features, removing outliers, and reducing noise. It is a traditional supervised learning algorithm. It is possible to customize the GB model to adapt feature-based noise elimination and improve signal quality. Since ECG signals are inherently nonstationary and time-dependent signal data, generated at high frequency (>= 360Hz), they consist of morphological features, like P-wave, T-Wave, and QRS complex, and it is too complex to preprocess the ECG data, due to the presence of powerline interference, motion artifacts, baseline wander, and electromyographic noise. Compared to traditional filtering methods, the customized GB can learn complex data patterns from the labeled dataset, separating signal from noise, and dynamically focus on ECG artifacts. It is explained mathematically below.

Let X be the input ECG signal segment $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ as a time-series data segment obtained from a lengthy ECG signal. $Y = \{y_1, y_2, \dots, y_i, \dots, y_n\}$ is a clean, reference, annotated ECG signal obtained from a clinical expert. The custom GB model is trained to reduce the loss value calculated between the predicted $\hat{y}_i$ and cleaned signals y_i, using the following formula:

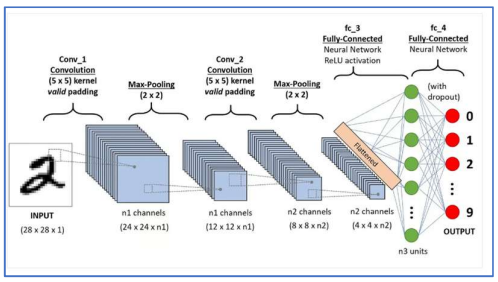
$$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i, \hat{y}_i)^2$$

This makes the custom GB model behave as a signal denoiser. The computational complexity is reduced by training the model on engineered features derived from ECG segments generated via sliding-window operations. Statistical, temporal, frequency-domain, and morphological features are certain features used.

CNN-based Image data pre-processing

The CNN is one of the deep learning algorithms, also known as ConvNet. It is primarily developed for image-based tasks, such as image classification, segmentation, and detection. CNNs learn spatial features through multiple layers, including convolutional

layers, ReLU activation functions, pooling layers, and fully connected layers. Convolutional layers extract spatial features from input data using filters, such as kernels. Activation functions are responsible for enabling the model to learn complex patterns by introducing non-linearity, and the ReLU activation function is the most commonly used. The pooling layers reduce the spatial dimensions of the feature maps to prevent overfitting, and the fully connected layers produce the final output predictions. CNNs are utilized in various applications due to their ability to process high-dimensional data. CNNs are more flexible, as they can adapt to time series or one-dimensional data across multiple domains.



CNN Architecture

In medical imaging, specifically Computed Tomography image data, the Convolutional Neural Network (CNN) model is more suitable. Before the CT medical images are fed into the CNN, preprocessing steps such as normalization, resampling, and cropping or padding are performed to ensure consistency and improve performance. The pre-processing steps of CNN are briefly explained below:

1. Normalization

Typically, CT images are stored in Hounsfield Units (HU), ranging from -1000 to +1000; however, it is possible to exceed these limits based on the scanning protocols. The range of the pixel values must be suitable for training, such as [0, 1] or [-1, 1], which can be achieved using CNN normalization. Let the 3D CT image voxel intensity of the location (x, y, z) is denoted as $I(x, y, z)$.

A common normalization function is mathematically represented as:

$$I_{norm(x,y,z)} = \frac{I(x,y,z) - \mu}{\sigma}$$

Or the min-max normalization is mathematically represented as:

$$I_{norm(x,y,z)} = \frac{I(x,y,z) - I_{min}}{I_{max} - I_{min}}$$

Where the voxel intensity in the volume of the mean and standard deviation is denoted as μ and σ. the I_{max}, I_{min} These are the intensity windowing limits for lung CT, with an HU range of [-1000, 400]. This step ensures that the CNN maintains a stable numerical range, prevents vanishing gradients, and facilitates faster convergence during training.

2. Resampling

Depending on the scanner and acquisition settings, the CT image may have various voxel spacings, including non-isotropic resolution. If the input of a CNN has uniform voxel spacing, such as 1 mm × 1 mm × 1 mm, then it performs well. The interpolating voxel values are used by resampling steps to transform the image into a fixed spacing. If the original spacing is (s_x, s_y, s_z) and the target is (t_x, t_y, t_z). The new image size is N_x, N_y, N_z is:

$$N_x = \left\lfloor \frac{s_x \cdot o_x}{t_x} \right\rfloor, \quad N_y = \left\lfloor \frac{s_y \cdot o_y}{t_y} \right\rfloor, \quad N_z = \left\lfloor \frac{s_z \cdot o_z}{t_z} \right\rfloor$$

Where the original image dimensions are denoted as $O_x, O_y \; and \; O_z$, and the CT volumes, usually, the interpolation is linear or spine-based. This step standardizes the anatomical proportions across the patients.

3. Cropping or padding

The CNN requires fixed dimensions, so after the sampling steps, input images may be larger or smaller than the required dimensions, such as 128 × 128 × 128. The cropping step extracts the ROI or the central-based patch:

$$I_{crop(x,y,z)} = I_{norm(x,y,z)}(x + x_0, y + y_0, z + z_0)$$

Where the central or ROI offset is denoted as x_0, x_y and x_z. If the input image dimensions are smaller than the target shape, then padding adds zero or mirror voxel values across all images:

$$I_{crop(x,y,z)} = \begin{cases} 0 & if\ outside\ bounds \\ I_{norm(x,y,z)} & otherwise \end{cases}$$

A standard input size is required for a consistent CNN-based application, and it is achieved by cropping or padding.

Result and Discussion

In this paper, three distinct types of data—image, signal, and time-series data—are collected from various datasets, and several pre-processing methods are applied to them. The primary aim of this paper is to demonstrate the efficacy of the deep learning model in accurately diagnosing diseases through image pre-processing techniques. As previously mentioned, the pre-processing techniques include normalization, rescaling, resampling, cropping, and noise reduction. To execute these steps, multiple learning algorithms and techniques are used, namely ML-based logistic regression (LR), linear discriminant analysis (LDA), support vector classifier (SVC), gradient boosting (GB), min-max preprocessing, and DL-based k-NN algorithms. The ML-based technique is employed to preprocess the input time-series data, while the DL-based technique is used to preprocess raw ECG signal and image data. The before-and-after effects of pre-processing, along with the prediction results for each data type, are discussed in detail in the following section.

Dataset

In this paper, three types of data are collected from three different publicly available datasets. The data [11] include various time-series data, encompassing records of patients with normal and abnormal heart disease. Similarly, the [12] consists of trained normal and abnormal ECG signal data. The [13] contains normal and abnormal abdominal image data. The attributes and other details of these datasets are discussed in the following subsections.

Experimental setup

The overall research work is conducted using various learning models. As mentioned, each model has a unique input data processing function. Based on the input data type, a suitable pre-processing step is performed. This entire simulation has been thoroughly tested, and the results were obtained using Python. The software is installed on a personal computer with 4 GB of RAM and an Intel Core i7-2630QM processor. It runs on Windows 10.

Time-Series Data Pre-processing using KNN Result

To evaluate the performance of the proposed model in processing numerical data, the input time series data are sourced from the publicly available Kaggle heart disease dataset [11]. Recently, heart disease has become one of the most serious health threats. It can result from high blood pressure, abnormal pulse rates, high cholesterol, and diabetes. Predicting heart disease and its severity is a crucial task in the medical field. Therefore, this paper applies various machine learning approaches to classify heart disease levels by preprocessing the input time-series data. The dataset comprises 920 rows and 16 columns, including patient ID, age, sex, location, chest pain type, blood pressure level, cholesterol level, blood sugar level, ECG result, heartbeat rate, blood disorder level, and additional variables. Each column provides different medical and personal details about the patients. The primary goal is to identify the total number of patients with and without heart disease. This is achieved through several preprocessing steps, including outlier detection, identification of missing values, replacement of null values, column renaming, removal of low-correlation features, feature scaling, and one-hot encoding. The following section discusses the results of these preprocessing steps on the time series data. From the overall results, a value of 0 indicates the absence of disease, while values of 1, 2, 3, and 4 represent the presence and type of disease.

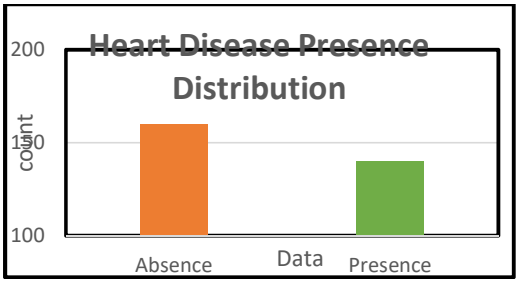


Figure 10: Heart Disease Presence Distribution

Figure-10 depicts the overall analysis results of the proposed model for detecting normal and heart disease patient data. The analysis results indicate that among the 299 data points, 160 are normal, and the remaining 139 are predicted to have diseases. Of these, 56, 35, 35, and 13 are expected to have heart disease types 1, 2, 3, and 4, respectively. The column renaming process enhances model efficiency once the total input data has been predicted.

Table-2 after removing the missing value

Country	Imputation Value	Total number of patients
Cleveland	297	304
Hungary	1	293
VA Long Beach	1	200
Switzerland	0	123

Table-2 illustrates the total number of missing values removed from the input data. It is

noted that the input dataset includes 920 missing values. After removing the missing values, only 299 values remain. It is evident from Table 2 that there are 297, 1, and 1 values observed from patients in Cleveland, VA, Long Beach, and Hungary, respectively. The figures also depict the total number of patients in each location identified after filling in the missing values. The analysis results show 304, 293, 200, and 123 patients from Cleveland, Hungary, VA Long Beach, and Switzerland, respectively.

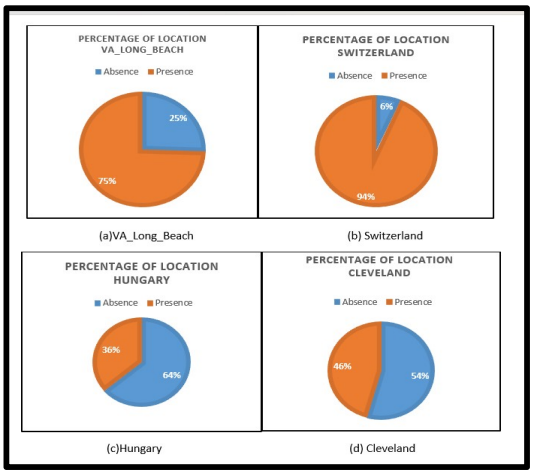


Figure 11: Total Number of Patients in Each Country Analyzed After Normalization

Now a 2×2 grid of subplots is created to preprocess and predict the total number of patients with normal and heart disease at each location. Based on the analysis result shown in Figure 11, the total number of patients with heart disease is predicted. That is, among 200 patients in VA_Long_Beach, 74.5% have heart disease; in Switzerland, 123 patients (93.5%) have heart disease; in Hungary, 36.2% of patients have heart disease; and in Cleveland, 45.7% of patients have heart disease.

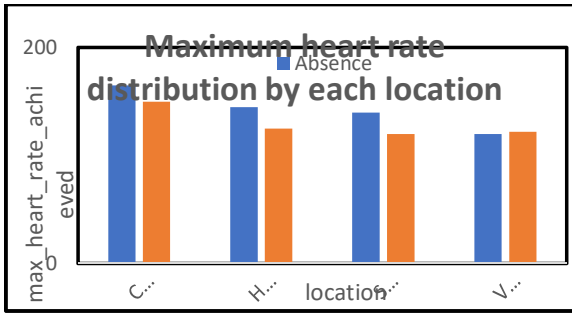


Figure-12. Maximum Heart Rate Distribution

Once the outlier-reduction process is complete, the age distribution ratios among heart disease patients across various locations are also identified. The predicted results are shown in Figure 12, and the results indicate that patients from VA Long Beach appear to be older than those from other locations. The results also indicate that heart rate decreases with age.

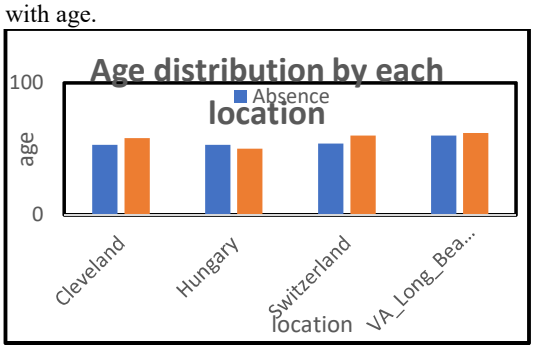


Figure-13: Age Distribution Ratio

After normalizing the input images, various features analyzed by the proposed model are also visualized. By scaling the input images, the maximum heart rates of patients across different locations are identified. The analysis results are presented in Figure 13. The maximum heart rate of patients with and without heart disease is analyzed and depicted in the figure. The analysis results indicate that patients in Cleveland have a higher maximum heart rate than those in other locations. In other words, compared to patients with heart disease, those without heart disease tend to have higher heart rates.

ECG Signal pre-processing using GB result

After preprocessing the input time-series data, the ECG signal is processed using the proposed wavelet transform. This method detects and removes artifacts in the raw signal. It improves the model's performance in detecting and diagnosing diseases from the input ECG signals. Two significant artifacts in the ECG signal are noise and baseline wander. Removing these artifacts from the raw signal is essential for achieving accurate results. Figure-14 shows the input raw ECG signal. The input ECG signals are collected from the publicly available Kaggle dataset “Shaoxing and Ningbo Hospital ECG database to evaluate the model's performance in removing artifacts from the ECG signals. " It includes 45,152 ECG recordings, each collected using a binary MATLAB v4 tool, with a duration of 10 seconds and a sampling frequency of 500 Hz [12].

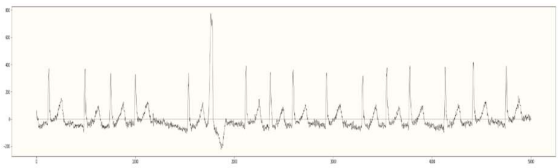


Figure-14 Raw Input ECG Signal

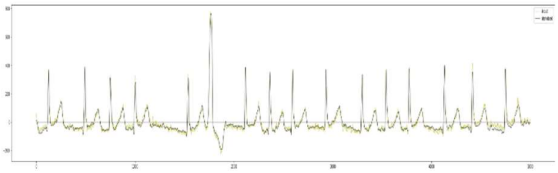


Figure-15 Before And After Denoised ECG Signal

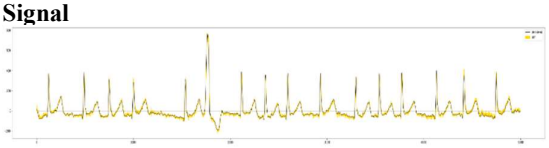


Figure-16: Denoised And Differentiated ECG Signal

Figure-15 shows the ECG signal analyzed using the proposed approach before and after preprocessing/denoising. The yellow in the image depicts the raw input signals with artifacts, while the black signal line indicates the denoised input signals achieved through the proposed wavelet transformer. Similarly, Figure-16 illustrates the difference between the input and denoised ECG signals, where the black line represents the denoised signal and the yellow line indicates the difference predicted by the proposed technique.

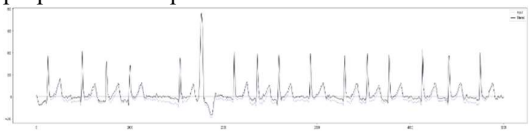


Figure-17: Before And After Filtered ECG Signal

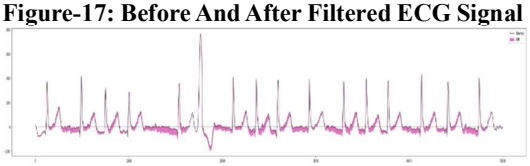


Figure-18 Differentiate Before And After Filtering Baseline

Then, by using a median filter, the baseline wander removal task is performed. The pink line in Figure-17 represents the input raw signal, while the black line represents the filtered ECG signal. Similarly, in Figure 18, the black line represents the denoised ECG signal, and the pink line indicates the difference between the raw and denoised signals.

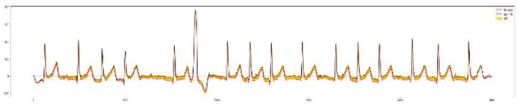
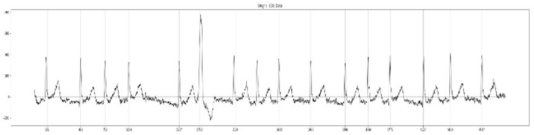
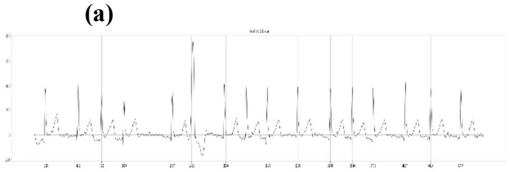


Figure-19 Denoising Comparison

Figure-19 depicts the best signal preprocessing step taken to remove artifacts from the raw input signals. The main objective of this evaluation is to determine which step needs to be performed first. For this, two sets of analyses are conducted: (i) denoising followed by filtering, and (ii) filtering followed by denoising. In Figure 19, the black, red, and yellow lines represent filtering ->denoising, denoising-> filtering, and the differences among the lines, respectively. The overall analysis suggests that denoising first, then filtering, yields better, more accurate results.



(a)



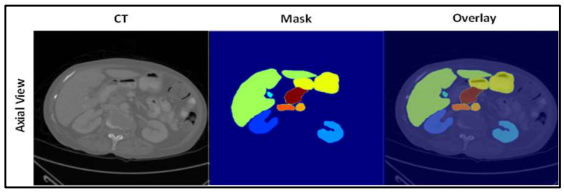
(b)

Figure-20 (A) Before (B) After Pre-Processing

Figure-20 depicts the final ECG signal obtained after performing the pre-processing step. To gain a clear understanding of Figure 20, both the raw and pre-processed final ECG signals are shown. The final ECG signal results show that the proposed wavelet transformer technique is more effective for denoising. At the same time, the median filter is more helpful in filtering baseline wander.

Image data pre-processing using the CNN result

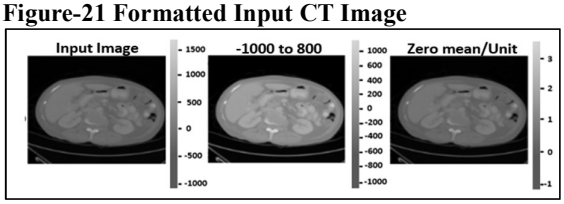
Now, image-based data is taken from the GitHub dataset and processed using deep learning techniques. As mentioned previously, the primary purpose of this research work is to preprocess the input image to produce more accurate classification and prediction results by following steps such as formatting Nifty data, normalization, resampling, cropping, and padding. To analyze the efficiency of the proposed model in preprocessing the image data, the input raw abdomen image data is taken from the multi-atlas labeling beyond the cranial vault workshop and challenge link [3,4]. It includes 50 abdominal CT scan reports. Figure-19 depicts the formatted Nifty data as an array of images. Figure-21 (a), (b), and (c) represent the input axial view CT image, masked abdomen image, and formatted overlay image, respectively.



(a) **(b)**

(c)

Figure-21 Formatted Input CT Image

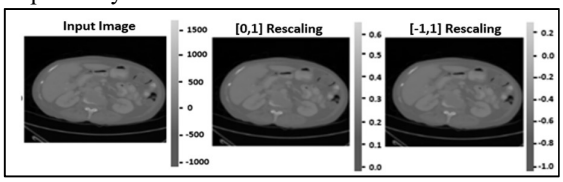


(a) (b)

(c)

Figure-22 Normalization Result

The image normalization is performed using the Min-Max technique. Figure-22 shows the normalized abdomen images and raw input images. Figure-22 (a), (b), and (c) depict the original, normalized from 1000 to 800, and the normalized CT image with the intensity of zero to three units, respectively.

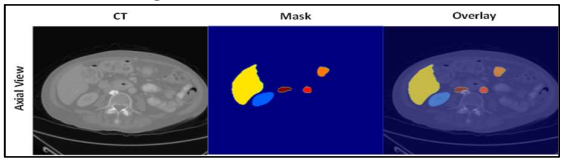


(a) (b)

(c)

Figure-23 Recycled Image

Similarly, Figure 23 illustrates the rescaled images produced by the proposed image pre-processing technique. Figure 23 (a) displays the raw input CT abdomen image, Figure 23 (b) showcases the [0,1] rescaled image, and Figure 23 (c) depicts the [-1,1] rescaled image. From these visuals, the proposed approach enables easy recognition and differentiation between infected and normal abdominal images.

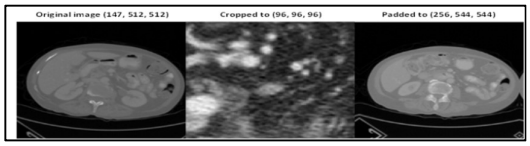


(a) (b)

(c)

Figure 24: Resampling Input CT Images

Furthermore, the image resampling step is used to resize or transform the input images into a different form. Generally, image resampling is performed more effectively using the KNN algorithm in the DL model. The input image is resampled by masking the affected area in the CT image. This is depicted in Figure 24, where the raw input CT image is shown in Figure 24(a), and the masked and resampled CT images are illustrated in Figures 24(b) and 24(c), respectively.



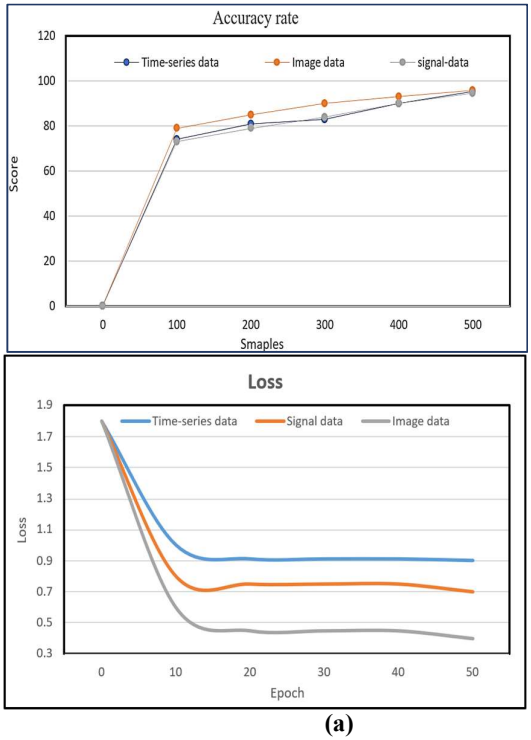
(a)

(b)

(c)

Figure-25 (A) Input CT Image, (B) Cropped Image, (C) Padded Image

Finally, the image cropping and padding step is performed to predict and diagnose the diseases in the input CT image. Figure-25 (a), (b), and (c) show the original, cropped, and padded input abdomen CT images. The analysis result shows. $147 \times 512 \times 512$ the stride input image is cropped to $96 \times 96 \times 96$. And padded to $256 \times 544 \times 544$. Figures 26 (a) and (b) depict the accuracy and loss rate of the proposed model for each data type. The overall analysis results emphasize that the proposed data pre-processing approach is more suitable for processing various types of input samples with a lower loss rate.



(a)

(b)

Figure 26 (A) Accuracy Rate (B) Loss of The Proposed Models

Table-1. Performance Comparison [ML & DL Models]

Models	Data Type	Accuracy	Precision	Recall	F1-score
KNN	Time-Series Data	0.97	0.98	0.98	0.98
GB	ECG signal Data	0.98	0.89	0.86	0.79
CNN	Image Data	0.99	0.97	0.98	0.97

The performance comparison results for the proposed ML and DL methods on preprocessed time-series data, ECG signals, and CT images are shown in Table 1. The analysis shows that the proposed KNN model achieved 97% accuracy, with precision, recall, and F1-score all at 0.98 for pre-processed input time-series data. The GB model achieved 0.98% accuracy on pre-processed ECG signal data, with precision 0.98, recall 0.89, and F1-score 0.79. The CNN model achieved 99% accuracy, with precision, recall, and F1-score rates of 0.97, 0.98, and 0.97, respectively. These comparison results demonstrate that the CNN model has a strong ability to accurately extract features from input images.

Table-2. Efficiency Comparison [ML & DL Models]

Models	Data Type	Latency Time (ms)	Computational Speed (inference/sec)	Error Rate (%)	Comments
KNN	Time-Series Data	120	45	3.0	During inference processing, it is slower.
GB	ECG signal Data	85	70	2.0	Performs better but has a slightly higher error rate due to the imbalanced ECG data.
CNN	Image Data	60	90	1.0	Performs better than others, with high speed, a low error rate, and low latency.

To further demonstrate the efficiency of the proposed ML and DL models in preprocessing various types of medical input data, the latency, computational speed, and error rate of the models are evaluated and presented in Table 2. The comparison results indicate that the KNN model processes the input time-series data at 45 inferences per second, with a latency of 120ms. The GB model preprocesses the input ECG signal data at a computational speed of 70 inferences per second, with a latency of 85ms and an error rate of 2.0%. Meanwhile, the CNN model has pre-processed the input image data with a latency of 60ms, an error rate of 1.0%, and a computational speed of 90 inferences per second. The comparison shows that, among the three models, the CNN-based image preprocessing method achieves lower latency and error rates while maintaining high computational speed.

Table-3. Accuracy Comparison concerning the Dataset

Models	Data Type	Accuracy Before pre-processing	Accuracy (after pre-processing)
KNN	Time-Series Data	0.89	0.97
GB	ECG signal Data	0.91	0.98
CNN	Image Data	0.93	0.99

The accuracy results of the proposed ML and DL models, before and after preprocessing, are presented in Table 3. The analysis indicates that the KNN model achieves 0.89% accuracy before pre-processing and reaches 0.97% after pre-processing. Similarly, the GB model achieves 0.91% accuracy before pre-processing and 0.98% after pre-processing. The CNN model achieves 0.93% accuracy before preprocessing the input image data and 0.99% after preprocessing. The results clearly show that the accuracy of all models improves both before and after pre-processing. In particular, the accuracy of the CNN model is comparatively high before and after pre-processing. This is because the multiple layers in the CNN model more accurately recognize patterns in the input image data, and applying pre-processing steps like resampling, cropping, or padding further improves the model's accuracy compared to other methods. The overall results illustrate that all the proposed models perform better with different types of medical data.

Conclusion

The main objective of this paper is to demonstrate the necessity of data preprocessing for medical datasets. The efficacy of data preprocessing is enhanced by using different methods for different tasks, such as noise removal, dimensionality reduction, transformations, and normalization. Preprocessing data improves data quality and the outcomes of data analytics. This paper employs efficient learning methods to uncover prior knowledge about the data and its analytical output. Predicting specific outcomes from a large medical dataset requires a high-cost computational model and expensive algorithms. The methods proposed in this paper are implemented in Python and tested on various datasets, such as healthcare records, medical images, and medical signals. The effectiveness of the data preprocessing methods is assessed by verifying the accuracy of the analysis. The results indicate that the preprocessing tasks reduce computational costs and complexity and enhance disease prediction accuracy.

Reference

1. Ampavathi, A. (2022). Research challenges and future directions for medical data processing. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 10(6), 633-652. https://doi.org/10.1080/21681163.2021.2018665
2. Shouval, R., Bondi, O., Mishan, H., Shimoni, A., Unger, R., & Nagler, A. (2014). Application of machine learning algorithms for clinical predictive modeling: a data-mining approach in SCT. Bone marrow transplantation, 49(3), 332-337. https://doi.org/10.1038/bmt.2013.146
3. Fatima, A., Nazir, N., & Khan, M. G. (2017). Data cleaning in data warehouse: A survey of data pre-processing techniques and tools. Int. J. Inf. Technol. Comput. Sci, 9(3), 50-61. DOI: 10.5815/ijitcs.2017.03.06
4. Safta, W., & Shaffie, A. (2024). Advancing pulmonary nodule diagnosis by integrating Engineered and Deep features extracted from CT scans. Algorithms, 17(4), 161. https://doi.org/10.3390/a17040161
5. Müller, H., & Freytag, J. C. (2005). Problems, methods, and challenges in comprehensive data cleansing (pp. 23-23). Professoren des Inst. Für Informatik.
6. Gudivada, V., Apon, A., & Ding, J. (2017). Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. International Journal on Advances in Software, 10(1), 1-20.
7. Silva, T. C., & Zhao, L. (2012). Network-based high-level data classification. IEEE Transactions on Neural Networks and Learning Systems, 23(6), 954-970. doi: 10.1109/TNNLS.2012.2195027.
8. Shantal, M., Othman, Z., & Bakar, A. A. (2023). A novel approach for data feature weighting using correlation coefficients and min–max normalization. Symmetry, 15(12), 2185. https://doi.org/10.3390/sym15122185
9. Khoei, T. T., & Singh, A. (2024). Data reduction in big data: a survey of methods, challenges and future directions. International Journal of Data Science and Analytics, 1-40.
10. Stapel, J. C., Hunnius, S., Bekkering, H., & Lindemann, O. (2015). The development of numerosity estimation: Evidence for a linear number representation early in life. Journal of Cognitive Psychology, 27(4), 400-412. https://doi.org/10.1080/20445911.2014.995668
11. https://www.kaggle.com/code/yujinsung/heart-disease-data-set/notebook.
12. https://www.kaggle.com/code/nelsonsharma/ecg-02-ecg-signal-pre-processing/input.
13. https://github.com/fitushar/3D-Medical-Imaging-Preprocessing-All-you-need/blob/master/README.md.
14. Kanani, P., & Padole, M. (2020). ECG heartbeat arrhythmia classification using time-series augmented signals and

deep learning approach. Procedia computer science, 171, 524-531. https://doi.org/10.1016/j.procs.2020.04.056

15. Nancy, A. A., Ravindran, D., Raj Vincent, P. D., Srinivasan, K., & Gutierrez Reina, D. (2022). Iot-cloud-based smart healthcare monitoring system for heart disease prediction via deep learning. Electronics, 11(15), 2292. https://doi.org/10.3390/electronics11152292
16. Bandyopadhyay, S., Bhaumik, A., & Poddar, S. (2021). Skin disease detection: machine learning vs deep learning. doi:10.20944/preprints 202109.0209.v1
17. Mishra, S., Dash, A., & Jena, L. (2021). Use of deep learning for disease detection and diagnosis. Bio-inspired neurocomputing, 181-201. https://doi.org/10.1007/978-981-15-5495-7_10
18. Roy, T. S., Roy, J. K., & Mandal, N. (2022). Classifier identification using deep learning and machine learning algorithms for the detection of valvular heart diseases. Biomedical Engineering Advances, 3, 100035. https://doi.org/10.1016/j.bea.2022.100035
19. Usman, S. M., Khalid, S., & Aslam, M. H. (2020). Epileptic seizures prediction using deep learning techniques. IEEE Access, 8, 39998-40007. DOI: 10.1109/ACCESS.2020.2976866
20. Gawhale, S., Upasani, D. E., Chaudhari, L., Khankal, D. V., Kumar, J. R. R., & Upadhye, V. A. (2023). EEG Signal Processing for the Identification of Sleeping Disorder Using Hybrid Deep Learning with Ensemble Machine Learning Classifier. International Journal of Intelligent Systems and Applications In Engineering, 11(10s), 113-129.
21. Alvi, A. M., Siuly, S., & Wang, H. (2021, October). Developing a deep learning-based approach for anomalies detection from EEG data. In International Conference on Web Information Systems Engineering (pp. 591-602). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-90888-1_45
22. Li, J. P., Haq, A. U., Din, S. U., Khan, J., Khan, A., & Saboor, A. (2020). Heart disease identification method using machine learning classification in e-healthcare. IEEE Access, 8, 107562-107582. Doi: 10.1109/ACCESS.2020.3001149.
23. Krori Dutta, K., Manohar, P., & Indira, K. (2023). Time and frequency domain pre-processing methods for multi-class seizure classification of epileptic EEG signals. Journal of Intelligent & Fuzzy Systems, (Preprint), 1-10. DOI: https://doi.org/10.70917/ijcisim-2025-0001
24. Debbal, S. M., & Hamza, C. (2021). Heart sounds analysis using the three wavelet transform versions: the continuous wavelet transform (CWT), the discrete wavelet transform (DWT), and the wavelet packet transform (PWT). *Journal of Cardiology Interventions*, *1*(1).
25. Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. *arXiv preprint arXiv:2305.17473*.