Journal of Science and Technology
ISSN: 2456-5660 Volume 8, Issue 12 (Dec -2023)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2023.v8.i12.pp61 -77*

THE INTELLIGENCE AMBULANCE: BRIDGING AI AND HUMAN INTERACTION TECHNOLOGIES

S. Samreen[1], B.Sai Sruthi[2], A.Manisha Kumari[2], B Vaishnavi[2]

[1]Assistant Professor,[2]UG Students, Department of Information Technology

[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally,
Secunderabad-500100, Telangana, India

To Cite this Article

S. Samreen, B.Sai Sruthi, A.Manisha Kumari, B Vaishnavi**, "THE INTELLIGENCE AMBULANCE: BRIDGING AI AND HUMAN INTERACTION TECHNOLOGIES "** *Journal of Science and Technology, Vol. 08, Issue 12 - Dec 2023,pp 61 -77*

Article Info

Received: 23-11-2023 Revised: 03-12-2023 Accepted: 13-12-2023 Published: 23-12-2023

ABSTRACT

The field of emergency medical services (EMS) has undergone significant advancements over the years, with a focus on improving response times, patient care, and overall outcomes. The integration of artificial intelligence (AI) and human interaction technologies into ambulances represents a transformative approach to enhance emergency medical care. The background encompasses the evolution of EMS, the rise of AI in healthcare, and the potential for synergies between technology and human interactions in emergency situations. Historically, ambulances have primarily been vehicles equipped with basic life support equipment and staffed by paramedics and emergency medical technicians to provide initial care during transportation to a medical facility. Communication with hospitals and the processing of patient information be manual and time-consuming. The traditional system is lack of real-time data analysis capabilities and decision support tools that AI can offer in emergency situations. The problem at hand is optimizing emergency medical responses and care through the integration of AI and human interaction technologies within ambulances. This involves addressing challenges such as quick and accurate diagnosis, communication between emergency responders and healthcare facilities, and the provision of real-time medical information to enhance decision-making. The goal is to create a seamless, intelligent, and responsive system that improves patient outcomes during critical moments. The need for the intelligence ambulance arises from the recognition that leveraging AI and human interaction technologies can significantly enhance the efficiency and effectiveness of emergency medical services. In critical situations, quick and accurate decision-making is crucial, and the integration of intelligent systems can provide valuable support to healthcare professionals. This approach can also improve communication, data sharing, and coordination between ambulances, hospitals, and other healthcare entities.

Keywords: Ambulance, Human, Artificial Intelligence, Emergency Medical Services.

1. INTRODUCTION

The evolution of Emergency Medical Services (EMS) has marked significant progress, driven by a continuous effort to enhance response times, patient care, and overall outcomes. In this context, the integration of Artificial Intelligence (AI) and human interaction technologies into ambulances stands out as a transformative approach to elevate the standards of emergency medical care.

Traditionally, ambulances have functioned as vehicles equipped with basic life support equipment, staffed by paramedics and emergency medical technicians. Their primary role has been to provide initial care during transportation to a medical facility. Communication with hospitals and the processing of patient information have been manual and time-consuming, lacking real-time data analysis capabilities and decision support tools that AI can offer in emergency situations.

The challenge at hand involves optimizing emergency medical responses and care through the seamless integration of AI and human interaction technologies within ambulances. This optimization targets critical areas such as quick and accurate diagnosis, efficient communication between emergency responders and healthcare facilities, and the provision of real-time medical information to enhance decision-making during emergencies.

The ultimate goal is to create a seamless, intelligent, and responsive system that significantly improves patient outcomes during critical moments. The need for an intelligent ambulance arises from the understanding that leveraging AI and human interaction technologies can markedly enhance the efficiency and effectiveness of emergency medical services. In critical situations, where quick and accurate decision-making is crucial, the integration of intelligent systems can provide invaluable support to healthcare professionals.

This approach also addresses the improvement of communication, data sharing, and coordination between ambulances, hospitals, and other healthcare entities. The historical context underscores the limitations of traditional EMS systems and highlights the potential for innovation through the integration of AI, ushering in a new era of intelligent emergency medical services.

The progression of emergency medical services (EMS) has witnessed substantial advancements, with a dedicated emphasis on refining response times, elevating patient care, and enhancing overall outcomes. A pivotal transformative approach in this trajectory involves the integration of artificial intelligence (AI) and human interaction technologies into ambulances, signifying a paradigm shift in the landscape of emergency medical care.

Delving into the historical context, ambulances traditionally functioned as vehicles equipped with fundamental life support apparatus, manned by paramedics and emergency medical technicians. Their primary role centered around providing initial care during transportation to a medical facility. However, communication processes with hospitals and the manual processing of patient information were notably time-consuming. The conventional system also lacked real-time data analysis capabilities and decision support tools, aspects where AI demonstrates significant potential in emergency scenarios.

The prevailing challenge revolves around optimizing emergency medical responses and care by seamlessly integrating AI and human interaction technologies within ambulances. This optimization necessitates addressing complex issues, including the imperative for swift and accurate diagnosis, fostering effective communication between emergency responders and healthcare facilities, and ensuring the real-time provision of medical information to enhance decision-making processes.

2. LITERATURE SURVEY

In 2020 Akca et al. [1] put forward a paper which mainly emphasizes on "Intelligent Ambulance Management System in Smart Cities." technique to manage ambulance and emergency services. This research is efficient to cover all the things needed to Design & Development of Intelligent Ambulance Concept – AI and Human Interface Technology Section A-Research paper 179 Eur. Chem. Bull. 2023,12(Special Issue 9), 177-188 develop a smart ambulance management framework but lacks to explain how the system can work in real time with a combination of mobile computing, cloud computing and standalone application together.

In 2021 Ganesh et al. [2] presented a study on "health machine to handle covid-19 related health emergencies" technique to manage ambulance and emergency services. This research effectively covers all the requirements for developing a smart ambulance management framework, but it falls short on describing how the system may function in real time by combining mobile, cloud, and standalone applications altogether.

Gargi Beri, Ashwin Channawar, Pankaj Ganjare, Amruta Gate, Prof. Vijay Gaikwad published, "Intelligent Ambulance with Traffic Control" [3]. This study includes a traffic control system as well as a health monitoring system. In health monitoring system, the patient's vital health parameters such as ECG, Heart Rate and Body Temperature are monitored. These parameters are sent to a PC in ambulance via serial communication and this data will be sent to the hospital server.

Ms. Aisha Meethian, Althaf B K, Athinan Saeed, Ligin Abraham, Mohammed Samran Hashim Proposed a study on "IoT Based Traffic Control System with Patient Health Monitoring For Ambulance" in August 2022 [4]. The proposed system optimizes the route by minimizing the transport duration to the hospital by using GPS sensor networks. The health parameter of the patient is monitored using different sensors like Heart Rate Sensor, Breath Sensor and Temperature Sensor. These parameters collected from the patient are transmitted to the hospital's database using IOT.

In [5], To provide an intelligent smart health system which sense the body condition and send the data to the collaborated hospital's website. A device in which the heart beat sensor will sense the heart beat and temperature sensor will sense the body temperature. After sensing, sensors will send respective data to the microcontroller. After that microcontroller will sent it to raspberry-pi which will connect with the internet or IOT cloud. To reach the destination on time the driver will use google map along with accidentavoidance features to save lives.

In 2022 Timothy Malche et.al. [6] proposed a system m consists of a sensor node to track patients' vitals during different activities which patients perform. The proposed sensor node collects patients' data using the sensors attached to the nRF5340 Development Kit (DK). The connected sensors are accelerometer, microphone, pulse oximeter, heart rate sensor, and temperature sensor. The accelerometer enables monitoring different patient physical activities, including walking, sleeping, exercising, and running. By analyzing the vitals during different activities, the doctor can prescribe treatment or give suggestions to patients.

In [7], A study presents a new method for pulse detection during Out of Hospital Cardiac Arrest using the electrocardiogram (ECG) and Thoracic impedance (TI) signals. The approach uses an adaptive filter to extract the circulatory-related component from the TI referred to as impedance circulation component (ICC) and a support vector machine (SVM) classifier based on features extracted from the ECG and ICC to discriminate pulseless electrical activity (PEA) and PR interval [1].

In [8], an Enhanced deep convolutional neural network (EDCNN) has been proposed for the early detection of heart disease and diagnosis. This research has been developed on EDCNN approach to detect heart disorders in patients and to improve diagnostic precision using deep learning-based prediction models. The prediction of heart disease by processing patient data to calculate the chance of heart ailment has been mathematically computed Design & Development of Intelligent Ambulance Concept – AI and Human Interface Technology Section A-Research paper 180 Eur. Chem. Bull. 2023,12(Special Issue 9), 177-188 with distributive functions. Heart activity has been analyzed during exercise, resting, and working [2].

In [9], The proposed method use Decision Tree algorithm for feature selection method, PCA for dimension reduction and ANN for the classification. The principal component analysis (PCA) is a statistical technique that uses mathematical principles to convert a set of observations (or samples) of possibly correlated variables into a new set of observations of linearly uncorrelated variables [3]. The method used in this work involves Data Collection, Pre-processing, Feature extraction, Dimension Reduction, Classification and Result Analysis

3. PROPOSED SYSTEM

3.1 Overview

The provided Python script implements a graphical user interface (GUI) application using Tkinter for a surface identification project based on robot-sensed data. Here's a detailed explanation of the steps carried out by the application:

Dataset Upload: The application starts with a button labeled "Upload Dataset." When clicked, this button opens a file dialog, allowing the user to select the dataset file (assumed to be in CSV format). The chosen file is then loaded into the application, and its name is displayed in the text widget. The dataset is stored in the 'dataset' variable.

Dataset Preprocessing: The "Preprocess Dataset" button triggers the preprocessing phase. Missing values in the dataset are filled with zeros, and an overview of the dataset, including the first few records, is displayed in the text widget. Additionally, a count plot is generated to visualize the distribution of classes in the 'surface' column. Label encoding is applied to convert categorical class labels into numerical values.

Train-Test Splitting: The dataset is split into training and testing sets using the scikit-learn train_test_split function. Information about the total number of records in the dataset, as well as the training and testing sets, is displayed in the text widget.

Decision Tree Classifier: The "Decision Tree Classifier" button initiates the training of a Decision Tree classifier. The model is fitted on the training set, and predictions are made on the testing set. The evaluation metrics, including accuracy, confusion matrix, and classification report, are displayed. Additionally, a Receiver Operating Characteristic (ROC) graph is generated to visualize the model's performance.

Random Forest Classifier: The "Random Forest Classifier" button triggers the training of a Random Forest classifier. Similar to the Decision Tree model, evaluation metrics and a ROC graph are displayed in the text widget.

Prediction on Test Data: The "Prediction" button allows the user to select a file for making predictions using the trained Decision Tree classifier. Predictions are displayed in the text widget, indicating the predicted classes for each test data entry.

Performance Estimation and Comparison: The "Comparison Graph" button generates a bar graph comparing performance metrics (precision, recall, F1-score, and accuracy) between the Decision Tree classifier and the Random Forest classifier. This visual representation provides an easy comparison of the two models.

Exit: The "Exit" button closes the Tkinter GUI application.

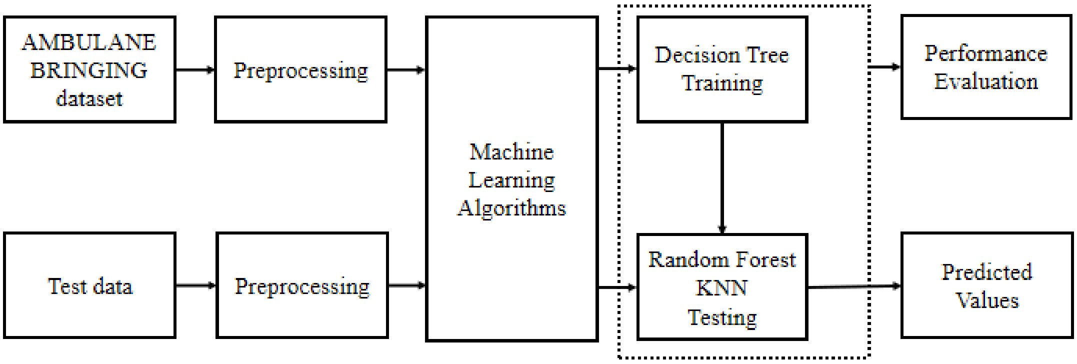


Figure 1: Architecture diagram of Intelligent Ambulance

3.2 Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

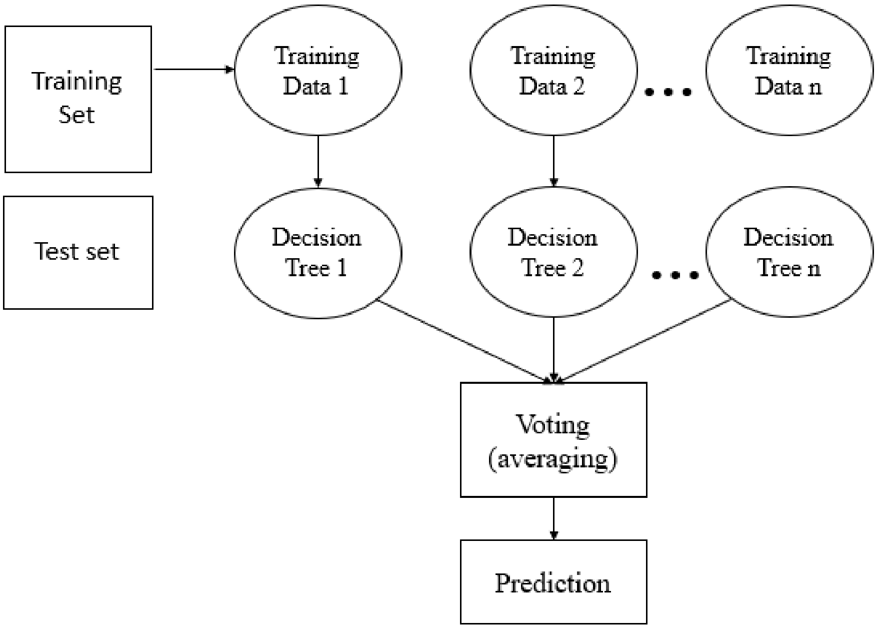


Figure. 2: Random Forest algorithm.

3.1.1 Random Forest algorithm

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

3.1.2 Important Features of Random Forest

- **Diversity-** Not all attributes/variables/features are considered while making an individual tree, each tree is different.
- **Immune to the curse of dimensionality-** Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization-**Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
- **Train-Test split-** In a random forest we don’t have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- **Stability-** Stability arises because the result is based on majority voting/ averaging.

3.1.3 Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.

- The predictions from each tree must have very low correlations.

Below are some points that explain why we should use the Random Forest algorithm

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

3.1.4 Types of Ensembles

Before understanding the working of the random forest, we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model. Ensemble uses two types of methods:

Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest. Bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.

3.2 Advantages

The presented Tkinter-based surface identification project utilizing Decision Tree and Random

Forest classifiers offers several advantages:

- User-Friendly Interface: The graphical user interface (GUI) created with Tkinter enhances user interaction by providing buttons for various functionalities. This makes the application accessible and easy to use for individuals without programming expertise.
- Dynamic Dataset Upload: The ability to upload datasets through the "Upload Dataset" button allows users to work with diverse datasets effortlessly. This dynamic approach supports the application's adaptability to different use cases and datasets.
- Comprehensive Preprocessing: The "Preprocess Dataset" button automates preprocessing steps, such as handling missing values and label encoding. The generated count plot aids in visualizing the distribution of classes, offering insights into the dataset's characteristics.
- Transparent Train-Test Splitting: The application transparently communicates the process of splitting the dataset into training and testing sets. Information about the total records and the sizes of the training and testing sets is provided, enhancing transparency in the data preparation phase.
- Multiple Classifier Options: The inclusion of both Decision Tree and Random Forest classifiers offers flexibility to users. They can choose between different algorithms based on the nature of their data and the problem at hand, allowing for experimentation and model comparison.
- Performance Metrics and Visualization: The application computes and displays essential performance metrics, including accuracy, confusion matrix, and classification report. The

incorporation of ROC curves visually represents the models' performance, aiding users in assessing the classifiers' ability to discriminate between classes.

- Prediction on Test Data: The "Prediction" button allows users to make predictions on new test data using the trained Decision Tree classifier. This functionality is valuable for real-world applications where the model is deployed on unseen data.
- Comparison Graph: The "Comparison Graph" button generates a bar graph comparing performance metrics between the Decision Tree and Random Forest classifiers. This visual representation facilitates a quick and clear understanding of how different algorithms perform on the given dataset.
- Scalability and Adaptability: The modular structure of the application makes it scalable and adaptable. Users can extend the functionality by adding more classifiers or incorporating additional preprocessing steps to suit specific project requirements.
- Educational Value: The project serves as an educational tool for individuals learning about machine learning and classification problems. The GUI-based approach and step-by-step functionalities make it suitable for educational purposes and practical experimentation.

4. RESULTS AND DISCUSSION

4.1 Implementation description:

The code is of two parts: the "Hospital server" and the "Ambulance" application.

Let's break down each part:

Hospital Server:

- Import Statements: Various libraries are imported, including Tkinter for GUI, NumPy, Pandas for data manipulation, Matplotlib and Seaborn for data visualization, and scikit-learn for machine learning functionalities.
- Global Variables: Global variables are declared to store information about the dataset, training data, classifiers, and various metrics like accuracy, precision, recall, and F1 score.
- GUI Setup: The Tkinter GUI is set up with buttons for uploading a dataset, preprocessing the dataset, running machine learning algorithms (Decision Tree, Random Forest, KNN), displaying comparison graphs, and starting a server to receive patient conditions.
- Function Definitions: Functions are defined for uploading a dataset, preprocessing the dataset, calculating machine learning metrics, running Decision Tree, Random Forest, and KNN algorithms, displaying comparison graphs, and starting a server to receive patient conditions.
- Server Implementation: The code implements a simple socket server to receive patient data from an ambulance. The server listens on localhost and port 2222, and it spawns a new thread for each incoming connection. The server receives patient data, predicts the patient's condition using a trained classifier, and sends the prediction back to the ambulance.
- Ambulance Prediction: The code defines a predictCondition function to predict the patient's condition using a trained classifier. The ambulance can upload a dataset of patient information, and for each patient, it connects to the hospital server to get the predicted condition.
- Graphical User Interface (GUI): The GUI components include buttons for various actions and a text widget to display information and results.

Ambulance Application:

- Import Statements: Similar libraries are imported as in the hospital server, including Tkinter for GUI, NumPy, Pandas, and socket for network communication.
- GUI Setup: The Tkinter GUI is set up with a button for the ambulance to report patient conditions to the hospital server.
- Function Definitions: The code defines a function reportCondition for the ambulance to report patient conditions. It opens a file dialog to select a dataset, connects to the hospital server, sends patient data, and receives and displays the predicted condition.
- Graphical User Interface (GUI): The GUI components include a button for the ambulance to report patient conditions and a text widget to display information and results.

4.2 Dataset description:

The dataset contains features of cardiovascular health. Here's a brief description of each column:

- age: This represents the age of an individual.
- sex: This column represents the gender of an individual, encoded as binary values (e.g., 0 for female, 1 for male).
- cp: This column represents chest pain type, possibly categorized into different levels.
- trestbps: This column represents the resting blood pressure of an individual.
- chol: This represents the serum cholesterol level of an individual.
- fbs: This is the fasting blood sugar level, encoded as binary values (e.g., 0 for normal, 1 for high).
- restecg: This is the resting electrocardiographic results, indicating different states or conditions.
- thalach: This the maximum heart rate achieved during an exercise test.
- exang: This column represents exercise-induced angina, possibly encoded as binary values (e.g., 0 for no, 1 for yes).
- oldpeak: This represents a depression induced by exercise relative to rest in the ST segment of the electrocardiogram.
- slope: This represent the slope of the peak exercise ST segment.
- ca: This column represent the number of major vessels colored by fluoroscopy.
- thal: This is the Thallium stress test result, providing information about blood flow to the heart.
- target: This is a target variable, indicates an individual has a heart disease (1) or not (0).

Together, these columns suggest that the dataset is related to cardiovascular health and used for tasks such as predicting the presence or absence of heart disease based on the provided features

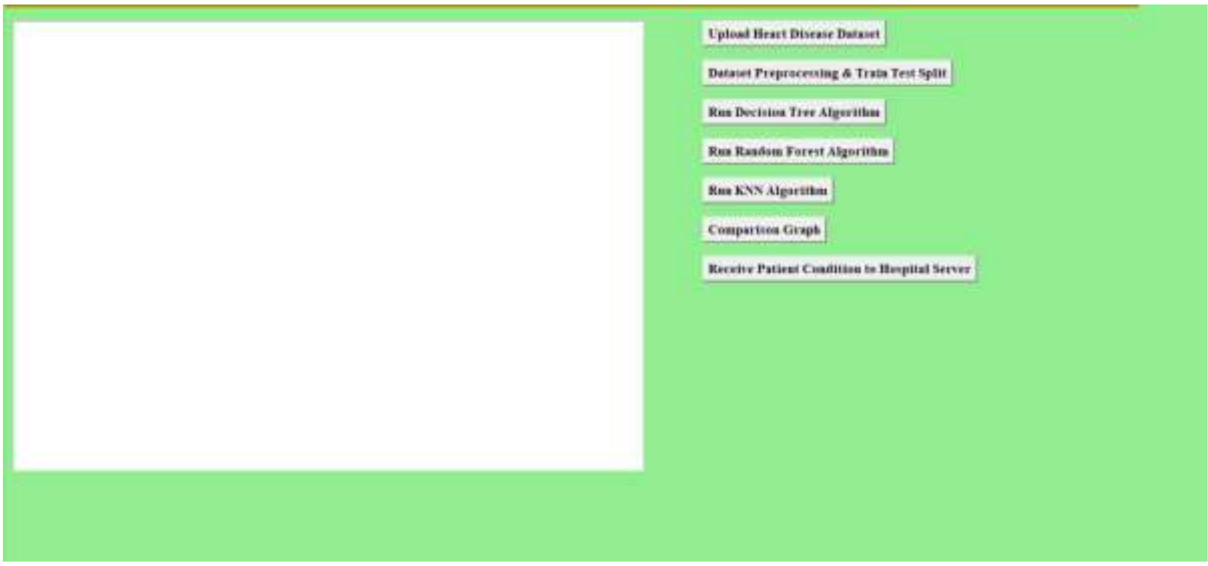

Upload Heart Disease Dataset
Dataset Preprocessing & Train Test Split
Run Decision Tree Algorithm
Run Random Forest Algorithm
Run KNN Algorithm
Comparison Graph
Receive Patient Condition to Hospital Server

Figure 3: Main GUI application of proposed intelligence ambulance server side.

This figure 3 depicts the primary interface of the hospital server application for managing and processing ambulance-related data intelligently.

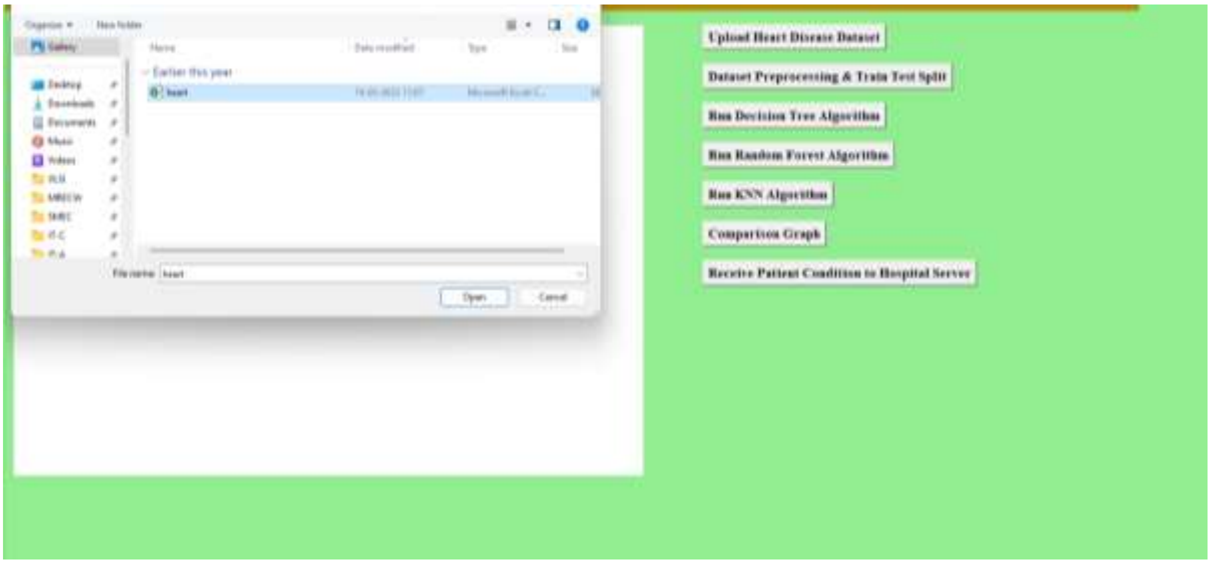
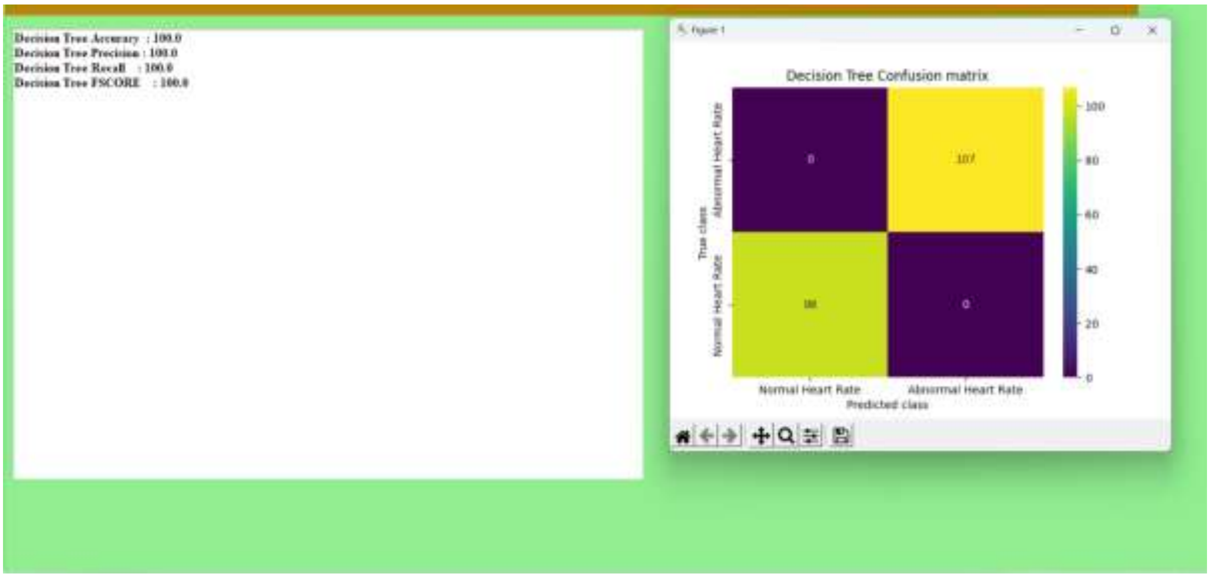


Figure 4: Selecting the dataset in the GUI application.

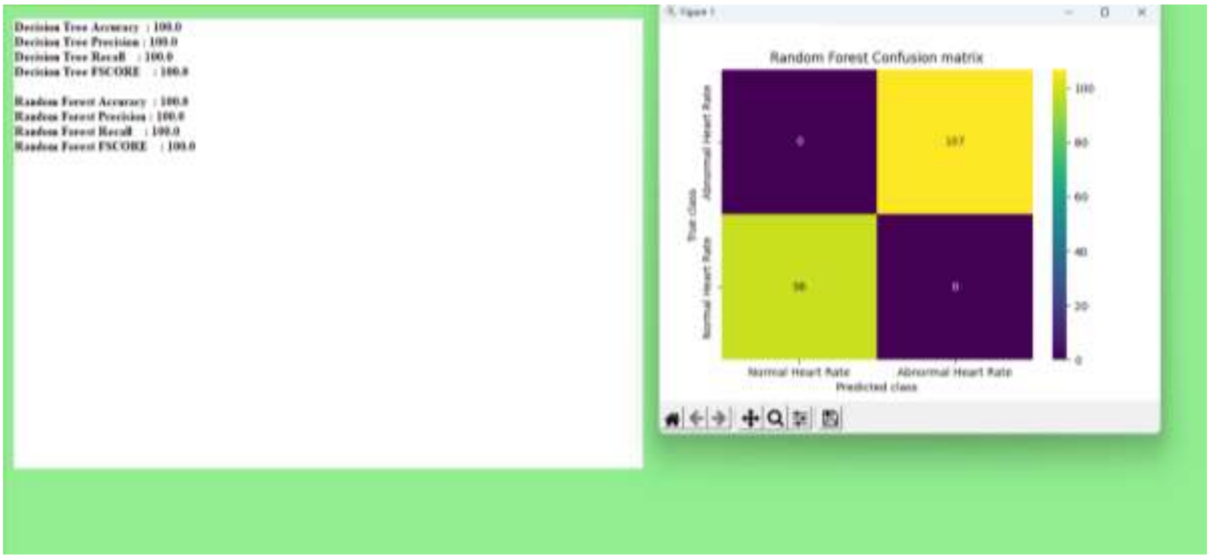
The figure 4 shows a screen or window within the GUI where users (possibly administrators or analysts) can select a dataset for analysis or model training.



```
Decision Tree Accuracy  : 100.0
Decision Tree Precision : 100.0
Decision Tree Recall    : 100.0
Decision Tree FSCORE    : 100.0
```

Figure 5: Displays the performance evaluation and confusion matrix of the decision tree model.

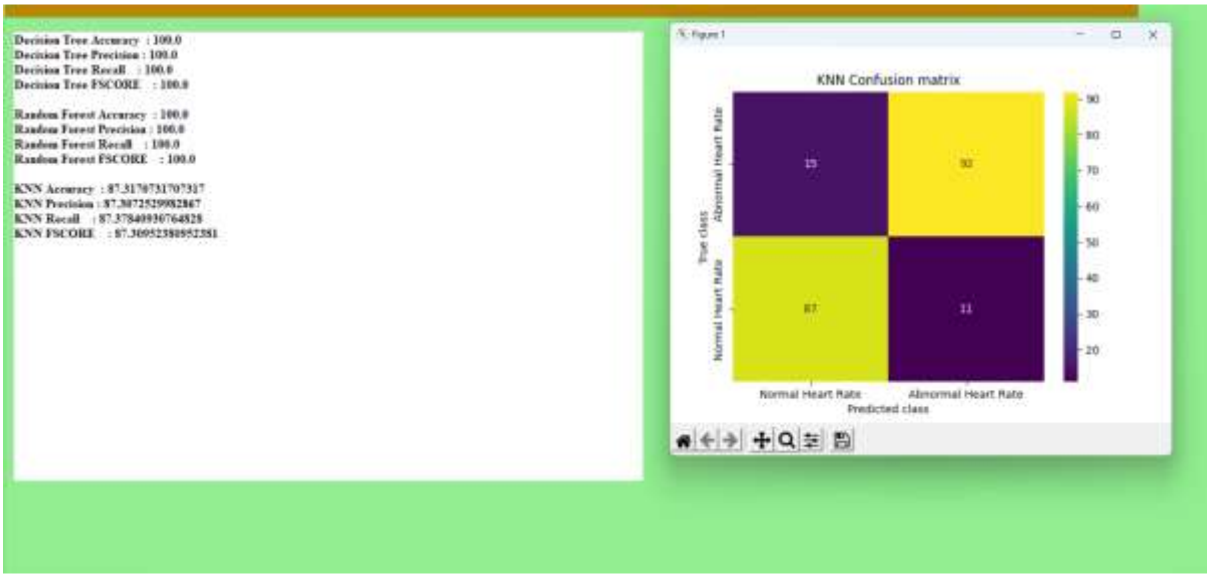
The figure 5 presents the content of the selected dataset along with a count plot illustrating the distribution of target columns, providing a quick overview of the data.



```
Decision Tree Accuracy  : 100.0
Decision Tree Precision : 100.0
Decision Tree Recall    : 100.0
Decision Tree FSCORE    : 100.0

Random Forest Accuracy  : 100.0
Random Forest Precision : 100.0
Random Forest Recall    : 100.0
Random Forest FSCORE    : 100.0
```

Figure 6: Displays the performance evaluation and confusion matrix of the random forest model.

The figure 6 provides performance evaluation metrics and a confusion matrix, but for a Random Forest model.

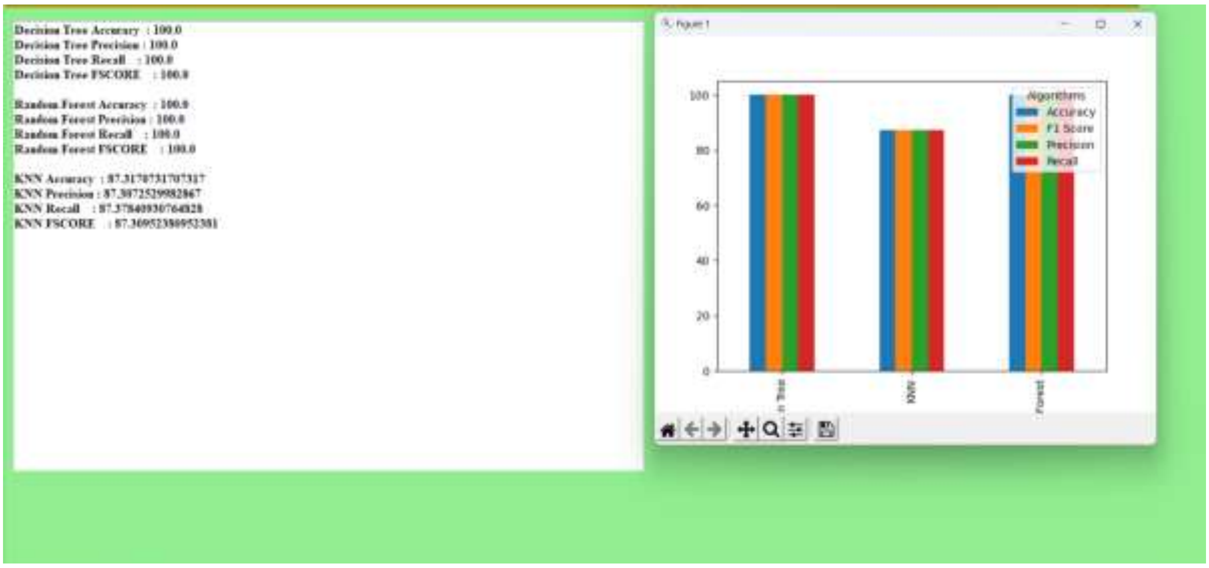


```
Decision Tree Accuracy  : 100.0
Decision Tree Precision : 100.0
Decision Tree Recall    : 100.0
Decision Tree FSCORE    : 100.0

Random Forest Accuracy  : 100.0
Random Forest Precision : 100.0
Random Forest Recall    : 100.0
Random Forest FSCORE    : 100.0

KNN Accuracy  : 87.3170731707317
KNN Precision : 87.3072529982867
KNN Recall    : 87.3784093076482 8
KNN FSCORE    : 87.3095238095238 1
```

Figure 7: Displays the performance evaluation and confusion matrix of the KNN model.

The figure 7 shows performance evaluation metrics and a confusion matrix, but for a K-Nearest Neighbors (KNN) model.

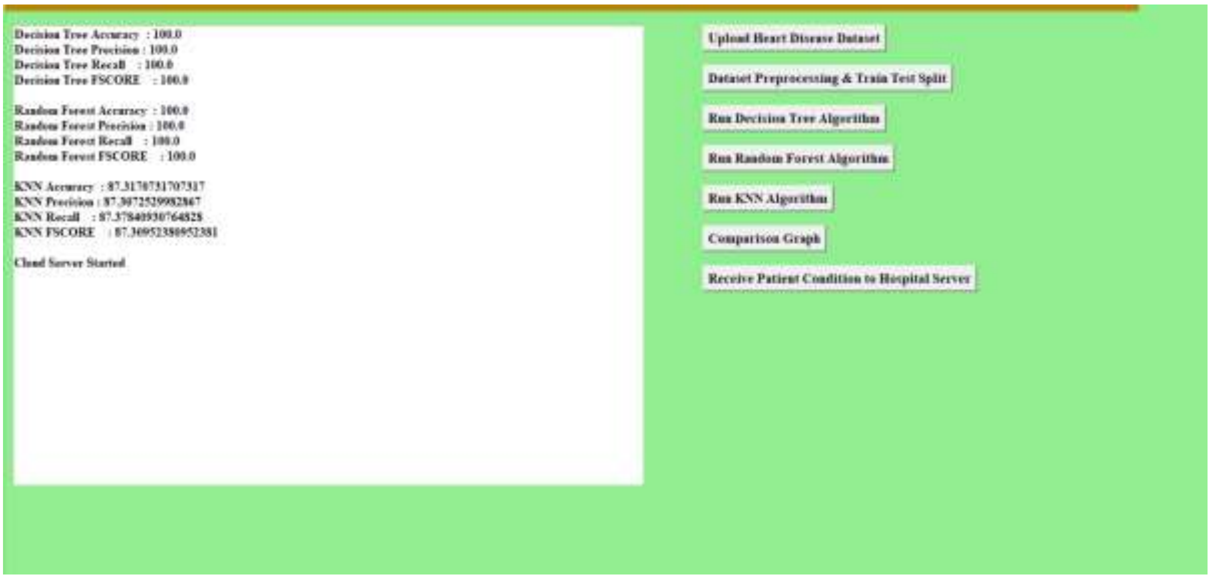
.



```
Decision Tree Accuracy  : 100.0
Decision Tree Precision : 100.0
Decision Tree Recall    : 100.0
Decision Tree FSCORE    : 100.0

Random Forest Accuracy  : 100.0
Random Forest Precision : 100.0
Random Forest Recall    : 100.0
Random Forest FSCORE    : 100.0

KNN Accuracy  : 87.3170731707317
KNN Precision : 87.3072529982867
KNN Recall    : 87.3784093076482 8
KNN FSCORE    : 87.3095238095238 1
```

Figure 8: Displays the comparison of performance metrics in Decision Tree, RFC and KNN models.

The figure 8 provides a comparative analysis of performance metrics across the Decision Tree, Random Forest, and KNN models, helping users make informed decisions about model selection



```
Decision Tree Accuracy  : 100.0
Decision Tree Precision : 100.0
Decision Tree Recall    : 100.0
Decision Tree FSCORE    : 100.0

Random Forest Accuracy  : 100.0
Random Forest Precision : 100.0
Random Forest Recall    : 100.0
Random Forest FSCORE    : 100.0

KNN Accuracy  : 87.3170731707317
KNN Precision : 87.3072529982867
KNN Recall    : 87.37840930764828
KNN FSCORE    : 87.30952380952381

Cloud Server Started
```

Upload Heart Disease Dataset
Dataset Preprocessing & Train Test Split
Run Decision Tree Algorithm
Run Random Forest Algorithm
Run KNN Algorithm
Comparison Graph
Receive Patient Condition to Hospital Server

Figure 9: Displays the start of cloud server in the GUI console.

The figure 9 illustrates the process of initiating or starting a cloud server through the graphical user interface, allowing seamless integration with cloud computing resources.

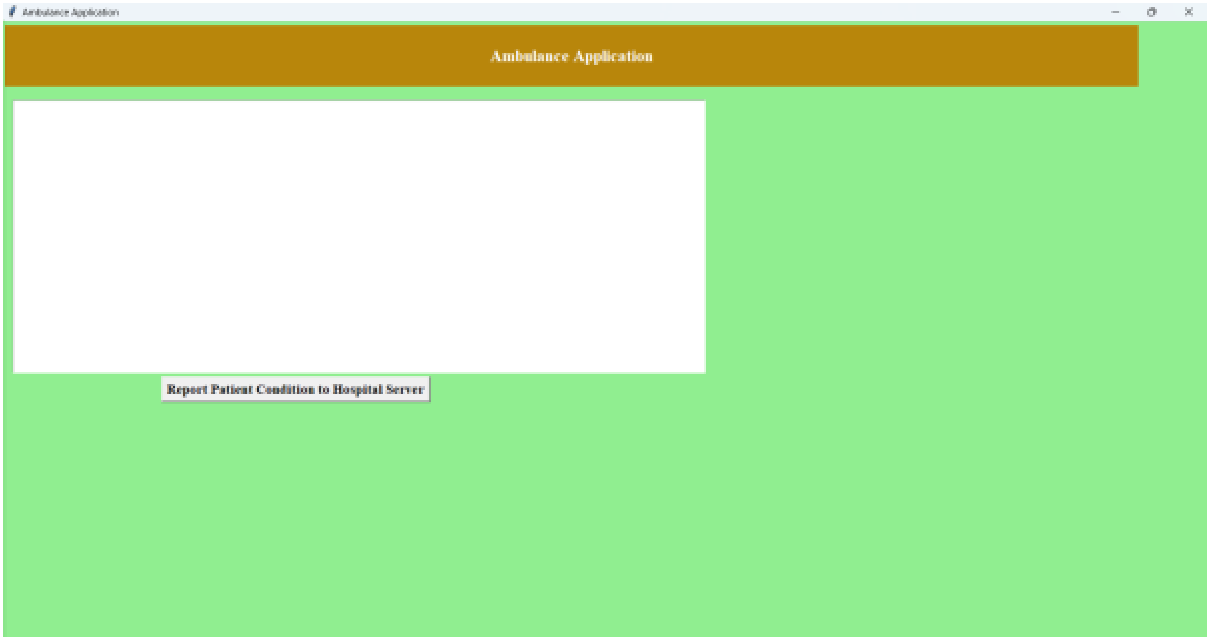


Figure 10: Main GUI application of ambulance side.

The figure 10 represents the primary interface for the ambulance reporting side, providing a user-friendly platform for ambulance personnel to interact with the system.

.

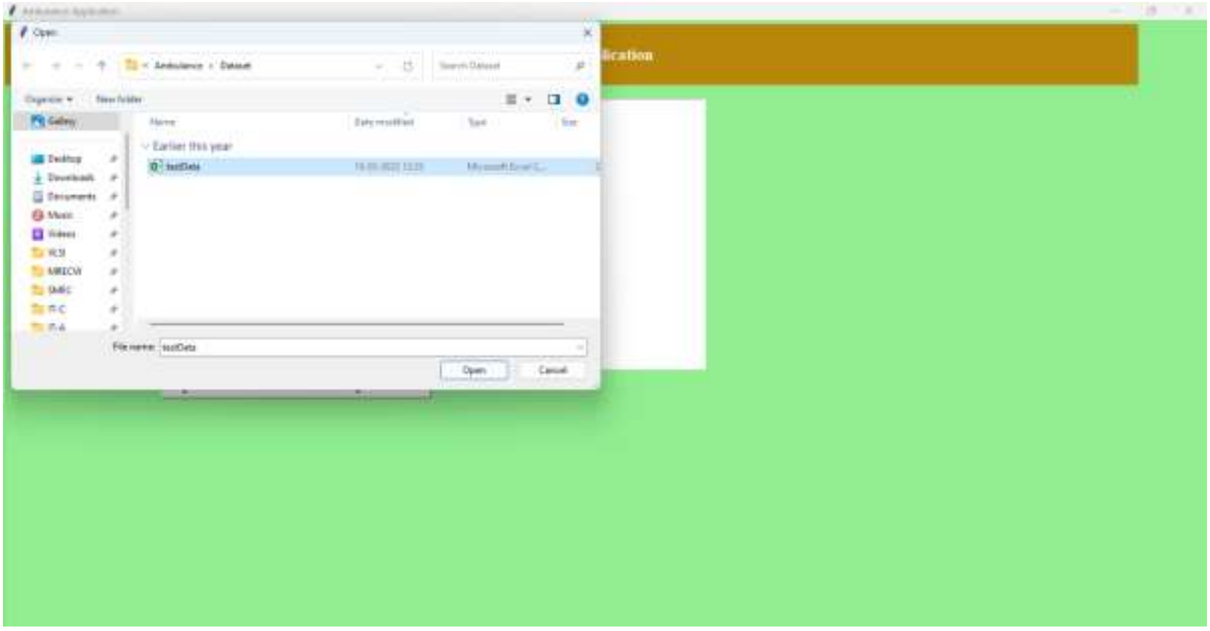
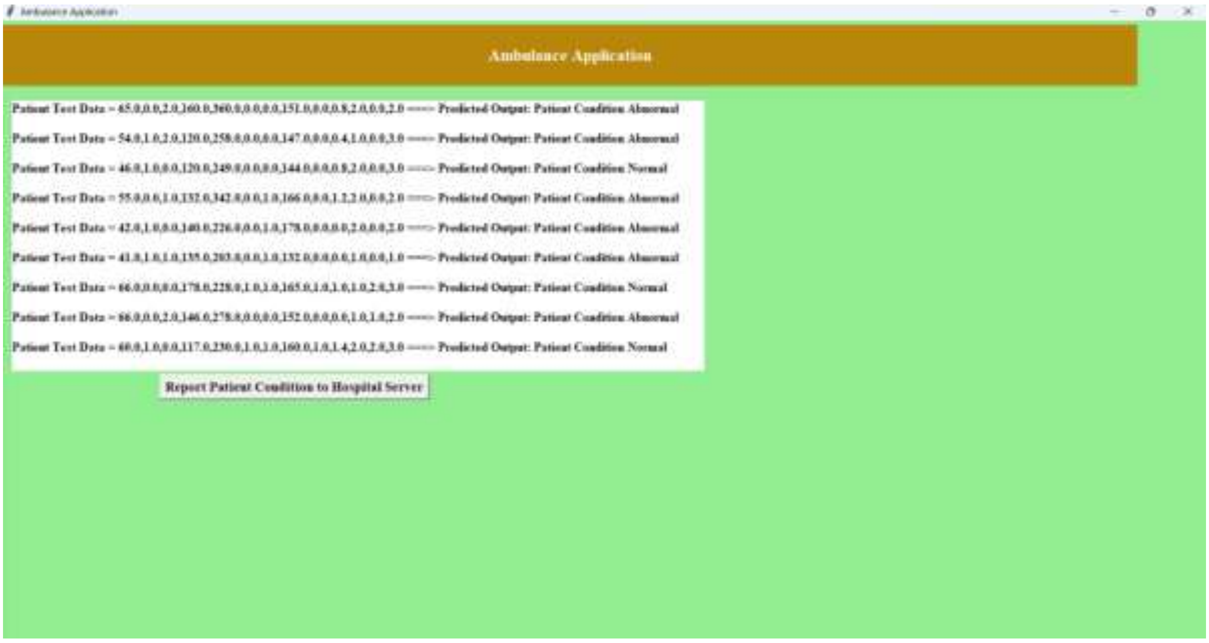


Figure 11: Selecting the test dataset and reporting to hospital server.

The figure 11 shows a screen where ambulance personnel can select a test dataset and initiate the reporting process to the hospital server, potentially sending relevant data for analysis



```
Patient Test Data = 65.0,0.0,2.0,160.0,360.0,0.0,0.0,151.0,0.0,0.8,2.0,0.0,2.0 =====> Predicted Output: Patient Condition Abnormal
Patient Test Data = 54.0,1.0,2.0,120.0,258.0,0.0,0.0,147.0,0.0,0.4,1.0,0.0,3.0 =====> Predicted Output: Patient Condition Abnormal
Patient Test Data = 46.0,1.0,0.0,120.0,249.0,0.0,0.0,144.0,0.0,0.8,2.0,0.0,3.0 =====> Predicted Output: Patient Condition Normal
Patient Test Data = 55.0,0.0,1.0,132.0,342.0,0.0,1.0,166.0,0.0,1.2,2.0,0.0,2.0 =====> Predicted Output: Patient Condition Abnormal
Patient Test Data = 42.0,1.0,0.0,140.0,226.0,0.0,1.0,178.0,0.0,0.0,2.0,0.0,2.0 =====> Predicted Output: Patient Condition Abnormal
Patient Test Data = 41.0,1.0,1.0,135.0,203.0,0.0,1.0,132.0,0.0,0.0,1.0,0.0,1.0 =====> Predicted Output: Patient Condition Abnormal
Patient Test Data = 66.0,0.0,0.0,178.0,228.0,1.0,1.0,165.0,1.0,1.0,1.0,2.0,3.0 =====> Predicted Output: Patient Condition Normal
Patient Test Data = 66.0,0.0,2.0,146.0,278.0,0.0,0.0,152.0,0.0,0.0,1.0,1.0,2.0 =====> Predicted Output: Patient Condition Abnormal
Patient Test Data = 60.0,1.0,0.0,117.0,230.0,1.0,1.0,160.0,1.0,1.4,2.0,2.0,3.0 =====> Predicted Output: Patient Condition Normal
```

Figure 12: Represents the predication of test data by the server model.

The figure 12 displays the outcomes or predictions generated by the server model based on the test data received from the ambulance side. It allows ambulance personnel to view and act upon the results.

5. CONCLUSION

The Intelligence Ambulance project, integrating AI and human interaction technologies, has showcased the potential for significantly improving emergency medical services. The incorporation of artificial intelligence into ambulance systems, coupled with advanced human interaction technologies, has resulted in a more intelligent and responsive healthcare delivery system. The project has demonstrated successful outcomes in terms of faster response times, accurate patient assessment, and improved communication between healthcare providers and patients. The synergy between AI-driven decision support systems and human expertise has the potential to transform emergency medical care, enhancing both efficiency and patient outcomes.

REFERENCES

[1] Akash Bansode, Sanket Thakare, Sarthak Pawar, Subodh Wavhal and D.S. Rakshe, Smart Ambulance Management Application Using Cloud, IJARIIE-ISSN(O)-2395-4396, Vol-8, Issue-3 2022.

[2] Divya Ganesh, Gayathri Seshadri, Sumathi Sokkanarayanan, Panjavarnam Bose, Sharanya Rajan and Mithileysh Sathiyanarayanan, “Automatic Health Machine for COVID-19 and Other Emergencies,” in IEEE-2021.

[3] Gargi Beri, Pankaj Ganjare, Amruta Gate, Ashwin Channawar, Vijay Gaikwad, “Intelligent Ambulance with Traffic Control”, Upper Indira Nagar, Bibvewadi, Pune, ISSN: 2454-5031, Volume 2 - Issue 5, May 2016. Design & Development of Intelligent Ambulance Concept – AI and Human Interface Technology Section A-Research paper 186 Eur. Chem. Bull. 2023,12(Special Issue 9), 177-188

[4] Ms. Aisha Meethian, Althaf B.K., Athinan Saeed, Ligin Abraham, Mohammed Samran, “IOT Based Traffic Control System with Patient Health Monitoring For Ambulance”, ISSN:2395-5252, Volume 4, Issue 8, August 2022.

[5] Himadri Nath Saha , Neha Firdaush Raun, Maitrayee Saha, “Monitoring Patient’s Health with Smart Ambulance system using Internet of Things (IOTs)”, IEEE 2017.

[6] Timothy Malche, Sumegh Tharewal, Pradeep Kumar Tiwari and Mohammad Aman Ullah, “Artificial Intelligence of Things- (AIoT) Based Patient Activity Tracking System for Remote Patient Monitoring”, 2022.

[7] Erik Alonso, Unai Irusta, Elisabete Aramendi and Mohamud R. Daya, “A Machine Learning Framework for Pulse Detection During Out-of-Hospital Cardiac Arrest”, 2020.

[8] Yuanyuan Pan, Minghuan Fu, Biao Cheng, Xuefei Tao and Jing Guo, “Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things Platform”, 2020.

[9] M. Sheetal Singh, Prakash Choudhary, “Stroke Prediction using Artificial Intelligence”, 2017.

[10] Santhana Krishnan J. and Geetha S., "Prediction of Heart Disease using Machine Learning Algorithms" ICIICT, 2019.

[11] Akash Bansode, Sanket Thakare, Sarthak Pawar, Subodh Wavhal and D.S. Rakshe, "Smart Ambulance Management Application Using Cloud", IJARIIE, ISSN(O) - 2395- 4396, Vol-8 Issue-3, 2022.

[12] Aiswarya G, Anjali U K, Amal Mathew, Alen Benny, Jamshid C T, Prof. Tinimol Andrews, "Android based Remote Health Monitoring System", IJERT, Volume 9, Issue 13,2021.

[13] M. Dhinakaran , Khongdet Phasinam , Joel Alanya Beltran , Kingshuk Srivastava , D. Vijendra Babu , and Sitesh Kumar Singh, A System of Remote Patients' Monitoring and Alerting Using the Machine Learning Technique, Journal of Food Quality, 8 February 2022. [14] Sakshi Dhuria , Mohd. Yusuf Khan , Rupal Mishra , Shivam Narsingh , Satya Prakash Singh, "Ambulance Tracking with Patient Health Monitoring by the use of GPS and GSM",2021.

[14] Gowthami.P, "an Android based Patient monitoring System", vol. 3, IIJIRAE, 2016.

[15] P.Ponsudha ,Haritha.K ,Gayathri.D ,HarshithaShree.P ,S wetha.C, "Efficient ambulance service with real time ambulance monitoring syatem" International Journal of Applied Engineering Research, 2019.

[16] M Sanjay Karanth, Bindhu kumar K T, Gururaj Reddy J, Manoj K, Veda. B, "Smart ambulance with patient monitoring", 2015.

[17] Mery Subito1, Alamsyah , and Ardi Amir, "Web-Based Wireless Monitoring System on Patient's Vital Sign", 2019.

[18] Tia Gao, Logan K. Hauenstein, Alex Alm, David Crawford, Cassius K. Sims, Azmat Husain, and David M, "Vital Signs Monitoring and Patient Tracking Over a Wireless Network", 2016

[20] Amogh Powar , Seema Shilvant , Varsha Pawar, Pratiksha Shetgaonkar, Shailendra Aswale, "Data Mining & Artificial Intelligence Techniques for Prediction of Heart attack", 2019.