**Machine Learning-Based Medicine Dispenser with Integrated App**

**An Engineering Project in Community Service**

## Phase – II Report Submitted by

**Team Members List**

|  |  |  |
| --- | --- | --- |
| 1 | Brishab Das | 22BCE10316 |
| 2 | Raunak Gupta | 22BCE11186 |
| 3 | Nikhil Appasaheb More | 22BCE11331 |
| 4 | Vivek kumar Mishra | 22BEY10099 |
| 5 | Milan P Samuel | 22BCG10175 |
| 6 | Aniruddha Pal | 22BCE11524 |
| 7 | Harshit Chimaniya | 22MEI10069 |
| 8 | Divyansh Rathore | 22MIM10022 |

***In partial fulfillment of the requirements for the degree of Bachelor of Technology***

******

# VIT Bhopal University Bhopal

**Madhya pradhesh - 466114**

**APRIL 2025**



**Bonafide Certificate**

Certified that this project report titled **"Machine Learning-Based Medicine Dispenser with Integrated App"** is the bonafide work of BRISHAB DAS 22BCE10316 , RAUNAK GUPTA 22BCE11186, NIKHIL APPASAHEB MORE 22BCE11331, ANIRUDDHA PAL 22BCE11524, MILAN P SAMUEL 22BCG10175, VIVEK KUMAR MISHRA 22BEY10099, HARSHIT CHIMANIYA 22MEI10069, DIVYANSH RATHORE 22MIM10022 who carried out the project work under my supervision.

This project report (Phase II) is submitted for the Project Viva-Voce examination held on…………………….

**Supervisor**

**Comment & Signature (Reviewer 1)**

**Comment & Signature (Reviewer 1I)**



**Declaration Of Originality**

We, hereby declare that this report entitled Machine Learning-Based Medicine Dispenser with Integrated App represents our original work carried out for the EPICS project as a student of VIT Bhopal University and, to the best of our knowledge, it contains no material previously published or written by another person, nor any material presented for the award of any other degree or diploma of VIT Bhopal University or any other institution. Works of other authors cited in this report have been duly acknowledged under the section "References".

Date Name & Reg No

10/04/25 Brishab Das 22BCE10316

10/04/25 Raunak Gupta 22BCE11186

10/04/25 Nikhil Appasaheb More 22BCE11331

10/04/25 Vivek Kumar Mishra 22BEY10099

10/04/25 Milan P Samuel 22BCG10175

10/04/25 Aniruddha Pal 22BCE11524

10/04/25 Harshit Chimaniya 22MEI10069

10/04/25 Divyansh Rathore 22MIM10022

**Acknowledgement**

First and foremost we would like to thank the Lord Almighty for his presence and immense blessings throughout the project work.

We wish to express our heartfelt gratitude to **Dr. Pushpinder Singh Patheja** (Dean, School of Computing Science and Engineering) and **Dr. Pon Harshavardhanan** ( Dean, School of Computing Science Engineering and Artificial Intelligence) for much of their valuable support encouragement in carrying out this work.

We would like to thank our internal guide **Dr. Jasmin Selvakumari Jeya I**, for continually guiding and actively participating in our project, giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of both the School of Computing Science and Engineering and School of Computing Science Engineering and Artificial Intelligence who extended directly or indirectly all support.

Last, but not the least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

**Abstract**

The healthcare industry is undergoing a transformative evolution with rapid advancements in technology, leading to innovative solutions aimed at enhancing patient outcomes. In this project, we propose a comprehensive system that combines **Machine Learning, IoT (Internet of Things)**, and App Development to address key challenges in medication management. The system centers around a **Machine Learning-Based Medicine Dispenser integrated with a mobile application**, offering an intelligent, automated, and user-friendly approach to healthcare delivery.

The core functionality involves analyzing symptoms provided by the user through the app using a machine learning model. This model predicts the most appropriate medicines for the symptoms and ensures accurate dispensing via an Arduino-controlled loT medicine dispenser. The hardware system is designed for precision and reliability, capable of dispensing medicines securely based on commands received through the app.

To enhance the user experience, the integrated app acts as an educational and interactive platform. It not only facilitates symptom input but also provides detailed information about the recommended medicine, including its intended use, potential side effects, and correct dosage instructions. This additional layer of information empowers users to make informed decisions about their health, bridging the gap between diagnosis and treatment.

This project represents a significant step toward addressing prevalent healthcare challenges, such as medication errors, dosage inaccuracies, and lack of accessibility to professional medical advice. By leveraging the strengths of machine learning, IoT, and app-based interfaces, the system offers an end-to-end solution that ensures efficiency, safety, and convenience for users. Its potential applications span hospitals, rural healthcare centers, and even home use, making it a versatile and scalable solution for modern healthcare needs.

**Index**

|  |  |  |
| --- | --- | --- |
| **S NO** | **TOPIC** | **PAGE NO** |
| **1**  **1.1**  **1.2**  **1.3** | Introduction  Motivation  Objective  Scope | **1**  **1**  **2**  **2** |
| **2**  **2.1**  **2.2**  **2.3**  **2.4**  **2.5**  **2.6**  **2.7** | Existing Work  Disease Prediction and Medicine Recommendation System  Medicine Recommendation System  IoT Based Smart Medicine Dispenser  Automated Dispensing System  24/7 General Medicine Vending Machine  A Prototype for Inteliigent Disease Prediction  Identified Shortcoming in Existing Framework | **3**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10** |
| **3**  **3.1**  **3.1.1**  **3.1.2**  **3.1.3**  **3.2**  **3.2.1**  **3.2.2**  **3.3**  **3.4** | Proposed Work  Architectural Framework  Overview  Hardware and Software Requirement  Mobile App Development  Working Principle  Model Traning and Validation  Dataset  Result and Discussion  Individual Contribution | **11**  **11**  **11**  **18**  **23**  **25**  **26**  **27**  **29**  **32** |
| **4** | Conclusion and Future Work | **40** |
| **5** | Reference | **41** |
| **6** | BioData | **44** |

**1. INTRODUCTION**

**1.1 Problem Statement**

Provision of accurate and timely medication is a highly critical component of quality healthcare; however, it remains a significant challenge in most areas, especially in remote and poorly resourced areas. Manual processes applied in medication management are prone to human error, expressed as incorrect dosages, missed doses, and misinterpretation of prescriptions. These issues may have serious health consequences, such as adverse drug reactions, ineffective treatment, and even hospitalization. Moreover, patients lack adequate knowledge about the medicines they consume, such as correct use possible side effects, and interactions with other drugs. On the other hand, healthcare providers are overwhelmed by the number of patients, and the complexities associated with modern medical treatments. The heavy workload may result in errors during the prescription and administration processes. The project outlined here seeks to solve such challenges by integrating Machine Learning, Internet of Things (IoT), and mobile application development into one intelligent system that automates dispensing medication while offering users with precious medical knowledge.

**1.2 Motivation**

The main driving force behind this project is the need to improve the accuracy, accessibility, and reliability of drug management systems. Trained doctors or pharmacists are not available in most rural or economically backward regions, and patients are left to self-administer their treatments without any guidance. Even in well-stocked city pharmacies and hospitals, healthcare workers are limited by extended working hours, heavy patient loads, and the natural likelihood of human error. Even more than that, patients suffering from chronic ailments or the elderly who take their medication daily may have difficulty adhering to a regime. This results in non-compliance, which in turn influences the outcome of treatment. Utilizing machine learning enables smart recommendations through personal health profiles, whereas smart devices like Arduino-controlled dispensers provide accurate delivery of medication. Having a mobile app available also streamlines the user interface by enabling reminders, tracking history of medication, and enhanced comprehension of prescriptions. In total, this project aims to close the gap between healthcare and technology to provide smarter, safer, and more accessible medical care.

**1.3 Objective**

* Use machine learning to automatically personalize medicine recommendations.
* Give full and clearly stated information regarding drugs, such as adverse effects, dosage instructions, and warning statements.
* Develop and design an Arduino-based dispenser that can properly dispense the correct amount at the correct time.
* Develop a user-friendly mobile application that integrates with the dispenser, enables monitoring, and delivers medication reminders and alerts.

**1.4 Scope**

The intelligent medicine dispensing system aids safe medication in rural areas lacking medical staff and reduces workload in urban hospitals. It supports elderly, chronically ill, and visually impaired patients by ensuring timely, accurate doses.

**2. EXISTING WORK/LITERATURE REVIEW**

This section provides an overview of the literature and technological developments in disease prediction, drug suggestion systems, drug dispensers based on IoT, and intelligent healthcare programs. It highlights the strengths, weaknesses, and research gap area of the present state of the art in the field that this project addresses through an integrated approach of machine learning, IoT, and mobile application development.

**2.1 Disease Prediction and Medicine Recommendation Systems**

Disease prediction and medicine suggestion systems are becoming more and more integral components in the development of intelligent healthcare systems. These systems apply machine learning algorithms to analyze patient data, identify potential diseases, and suggest appropriate medicines based on learned patterns [1]. Different algorithms have been employed for this, including Decision Trees, Random Forests, and deep learning algorithms like Convolutional Neural Networks (CNNs) [13]. Each algorithm represents a unique compromise among predictive performance, interpretability, and efficiency.

Though CNNs and neural network-based models generally deliver strong accuracy levels when it comes to classification, they are commonly opaque and hungry for data and computation [16]. Conversely, Decision Tree models provide a clearer and more explainable architecture, which is especially vital within the healthcare environment where knowing the logic behind a recommendation is as valuable as the recommendation itself [5]. Decision Trees give a visual, branching sequence of logic that allows easier tracking by developers and healthcare providers alike of how a given output was arrived at. This property of Decision Trees makes them particularly well-suited for application in fields where interpretability is a must [13]. In many medical cases, physicians are required to understand and validate the suggestions provided by AI systems. An interpretable model such as a Decision Tree can show exactly which symptoms or features led to a particular diagnosis or medication suggestion [6]. This establishes trust in the system and enables clinicians to make clinical decisions on the basis of both algorithmic suggestions and their own clinical judgment.

For the use of our system, the Decision Tree Classifier was selected as the base algorithm due to these specific reasons [8]. It is appropriate for the nature of medical data since it is usually tabular with well-defined features such as symptoms, age, gender, or history. The model can efficiently process this data, identify suitable patterns, and give recommendations in real-time. Another major benefit of using Decision Trees is their lightweight computational requirement [17]. In contrast to more sophisticated models, they don't require high-performance computing resources, and as such, they can be deployed in mobile health applications, low-resource environments, or embedded systems like intelligent dispensers. Their fast training time and prediction speed make them enable the responsiveness required in real-time healthcare systems. Moreover, reproducibility and consistency of Decision Tree models result in reliability in medical decision-making [20]. Given the same input, the model always follows the same logical paths, and that's a great thing for mission-critical tasks. It also makes it simple to test, validate, and maintain the system, and thus the system is easier to change or improve based on new medical knowledge becoming available.

In conclusion, Decision Tree Classifier is a good foundation for a disease forecasting and medicine recommendation system [13]. Its simplicity, efficiency, and interpretability qualify it as a suitable candidate for healthcare systems where transparency and trust are paramount. This will ensure that the system will not only be efficient but will also help clinicians give accurate and customized treatment to patients.

**2.2 Medicine Recommendation System**

Medicine suggestion systems are turning into an important part of the modern healthcare technology, especially due to the increased need to mechanize diagnosis and treatment suggestion [13]. One of the most important features of these systems is their ability to accurately parse and process inputs from users. In most cases, users describe their symptoms using unstructured natural language in terms of sentences of varying shape and vocabulary. This is a big challenge to machine learning algorithms, which require structured, numeric input. To bridge this deficiency, Natural Language Processing (NLP) is employed to convert free-text symptom descriptions to computational-ready data [5].

In our system, NLP plays the main role of processing and interpreting symptoms entered by patients. When users enter symptoms in free text, the system begins with text preprocessing techniques. This is initiated with text normalization, in which the input is cleaned by making everything lowercase, removing punctuation, and spelling correction if needed. This helps ensure uniformity in inputs such that the same symptom is not entered differently [9].

Upon normalization, tokenization is carried out. Tokenization splits the input text into tokens or words that are used as the basis units for the latter analysis. After tokenizing the input text, the other major step following that is vectorization—the process converting text information to numerical vectors machineries for processing machine learning algorithms. Our system makes use of mechanisms such as CountVectorizer and Term Frequency-Inverse Document Frequency (TF-IDF) for it [6]. CountVectorizer transforms the tokenized input into a word frequency vector that describes how many times each term appears in the document. TF-IDF, however, normalizes these counts with respect to the importance of each word relative to its frequency in a larger universe of text. This makes sure that the system focuses on more informative and unique terms rather than common words that may not be very valuable to classification accuracy [12]. Once the textual symptom data has been vectorized, it becomes machine-processable. The machine learning model then processes the structured input and assigns it to known patterns of symptoms and corresponding medicines [14]. This pipeline allows the system to function properly even with highly variable human input, thereby improving its flexibility and usability. Adding NLP enhances the system's usability. The users are not required to select symptoms from a list or use medical jargon— they can simply describe how they are feeling in their own words. This lowers the barrier to entry for lay users and enhances the application's usability in the field, especially for users with little technical or medical knowledge [8].

Moreover, converting qualitative descriptions to quantitative data improves the overall performance of the classifier. It translates vague or imprecise symptom reports into structured inputs and allows the system to make more precise recommendations. It reduces misclassification and maximizes the relevance of suggested medications [11].

In general, NLP is one of the main elements of our system for suggesting medicine. It ensures accurate interpretation, processing, and conversion of users' unstructured symptom input into an analyzable format. This not only improves the performance of the model but also improves the user experience because natural, flexible interaction with the system is supported [10].

1. **3 IoT-Based Smart Medicine Dispenser**

Application of the Internet of Things (IoT) in medicine has brought forth new opportunities for innovation, particularly in terms of automated drug dispensing [3]. Smart medicine dispensers apply interconnectivity technologies to enhance drug adherence, reduce human error, and provide remote access to essential health-related information [22]. Such dispensing systems typically incorporate a combination of sensors, microcontrollers, actuators, and communication modules to sense, schedule, and regulate the dispensing of medications to patients [23].

In our project, the concept of a smart, IoT-driven medicine dispenser was the driving force behind defining the design and functionality of the system. Inspired by previous research in this area, we developed our dispenser around an Arduino-based microcontroller, which is the heart of the system [24]. The microcontroller is connected to certain hardware components such as servo motors for mechanical dispensing, Real-Time Clock (RTC) for scheduling in real-time, and Wi-Fi modules for connectivity to external devices and cloud services [21]. Real-time monitoring and enforcement of schedule is one of the major conclusions drawn from the available technologies in IoT-based dispensers [19]. The RTC module allows for medication administration at exactly the scheduled times, reducing the chances of delayed or missed doses. The servo motors are engineered to open the programmed compartments at timed intervals, delivering the appropriate medication to the patient with precision and reliability [2]. This level of high automation is intended to deliver higher user safety and less human intervention. In addition, IoT connectivity is essential to enable features such as remote monitoring, alerts, and notifications [25]. By connecting with Wi-Fi modules, the dispenser can give real-time feedback to caregivers or family members regarding whether the user has consumed their medication. In the event of a missed or delayed dose, an alert can be sent immediately to a connected device such as a smartphone or computer [18]. This not only increases drug compliance but also adds a sense of responsibility and concern, particularly for elderly or chronically ill patients who may require assistance.

Our project further leverages cloud connectivity in storing users' data in a secure manner, such as medication regimens, user interactions, and compliance data [7]. The information is accessible for examination by healthcare practitioners or caretakers to gauge patient behavior, improve drug plans, or discern patterns that may signify certain underlying problems. Centralized data storage closes the information gap between patients and practitioners, facilitating better decision-making and anticipatory healthcare measures [20]. Another advantage of implementing IoT in our dispensing system is the scalability and flexibility of the solution [1]. New functionality can be added through software updates, and existing components can be upgraded without redesigning the whole system. Moreover, remote access allows administrators to change medication schedules or system settings remotely, giving users and medical staff more overall convenience and control [16].

In conclusion, IoT has also played a pivotal role in designing our smart medicine dispenser by contributing tools and infrastructures for live interaction, automaton, as well as faraway healthcare intervention. By utilization of microcontrollers, sensors, and communication modules, our design aims to deliver a modern, effective, as well as people-centered solution in the management of medication [23].

**2.4 Automated Dispensing System**

Automated drug dispensing systems are a significant leap in health technology, offering solutions that can dispense medicine without constant human supervision [2]. They are particularly well-suited in environments where the presence of pharmacists or medical personnel can be minimal. Such a system, described in earlier developments, utilizes simple input interfaces to enable users to have direct access to medication through a vending machine-type of setup [19]. Though it does not involve integrated machine learning or diagnostic capability, it does demonstrate the possibility and benefit of automated medicine distribution by utilizing basic hardware and software components.

This concept has had a direct impact on informing the hardware of our own medicine dispensing system. Specifically, the use of servo motors and Arduino control has been directly informed by known methods of achieving precision in medicine delivery [24]. Servo motors drive the movement of compartments or trays so that the correct quantity is dispensed with minimal variation. The Arduino microcontroller provides the logic and timing necessary to carry out dispensing operations according to user input or scheduled events [21]. Moreover, research into such systems emphasized key areas like error handling and user authentication [25]. These are key to providing safety, particularly in systems that operate autonomously. In our design, error detection features are built in to avoid improper dosing or hardware faults, e.g., jammed compartments or faulty motor operations. In addition, user authentication procedures—e.g., password entry or RFID scanning—have been integrated to ensure that only valid users can access the medication [4].

Real-time monitoring and control also form an important component of the system. The ability to respond promptly to user activity or system failure reinforces reliability and fosters user trust [22]. Overall, all lessons learned from past automated dispensing systems have played a vital role in designing a safe, responsive, and user-friendly drug delivery system.

**2.5 24/7 General Medicine Vending Machine**

The idea of a 24/7 general medicine vending machine presents a new, high-technology option for users to obtain necessary drugs at any time without the intervention of human beings. Such systems are made to be autonomous, typically located in public spaces like transportation terminals, community centers, or rural health centers. They provide users with rapid and effective access to over-the-counter drugs, and in some cases, prescription drugs, through a mix of secure interfaces and smart inventory management.

Some of the shared features of these machines include GPS tracking, where users and system administrators can track the machine location and usage in real-time. It is especially useful in the event of mobile or temporary installations in rural or underserved areas. They also include secure payment terminals that support various modes of payments, including card payments, mobile wallets, and contactless payments. It is not only user-friendly but also promotes accountability and transparency of finances. Automated stock management is another significant feature of such systems that keeps stock levels and sends reminder messages for restock when these levels go below a certain threshold. This minimizes human interference and provides in-demand medicines at all times.

These aspects can be incorporated directly into our own future roadmap for our smart medicine dispenser. While the current focus is on personalized medicine dispensing and suggestion based on symptoms, features such as secure digital payment and GPS location tracking are part of our system's roadmap. These aspects would facilitate the dispenser being deployed in rural or semi-urban regions, where 24/7 access to medication is a question of life and death. Through expansion of the functionality beyond mere dispensing, our project will transform the system into a completely independent, smart healthcare kiosk that maximizes accessibility, dependability, and ease of use for users across a wide range of settings.

**2.6 A Prototype for Intelligent Disease Prediction and Pharmaceutical Recommendation**

The idea of a 24/7 general medicine vending machine presents a new, high-technology option for users to obtain necessary drugs at any time without the intervention of human beings [23]. Such systems are made to be autonomous, typically located in public spaces like transportation terminals, community centers, or rural health centers. They provide users with rapid and effective access to over-the-counter drugs, and in some cases, prescription drugs, through a mix of secure interfaces and smart inventory management.

Some of the shared features of these machines include GPS tracking, where users and system administrators can track the machine location and usage in real-time [21]. It is especially useful in the event of mobile or temporary installations in rural or underserved areas. They also include secure payment terminals that support various modes of payments, including card payments, mobile wallets, and contactless payments [4]. It is not only user-friendly but also promotes accountability and transparency of finances. Automated stock management is another significant feature of such systems that keeps stock levels and sends reminder messages for restock when these levels go below a certain threshold [22]. This minimizes human interference and provides in-demand medicines at all times.

These aspects can be incorporated directly into our own future roadmap for our smart medicine dispenser. While the current focus is on personalized medicine dispensing and suggestion based on symptoms, features such as secure digital payment and GPS location tracking are part of our system's roadmap. These aspects would facilitate the dispenser being deployed in rural or semi-urban regions, where 24/7 access to medication is a question of life and death. Through expansion of the functionality beyond mere dispensing, our project will transform the system into a completely independent, smart healthcare kiosk that maximizes accessibility, dependability, and ease of use for users across a wide range of settings.

**2.7 Identified Shortcomings in Existing Frameworks**

Despite the advancements made in health technology, including machine learning-based diagnosis, Internet of Things-based dispensers, and self-service medication venders, there are certain intrinsic limitations in current implementations [2]. They are isolated solutions that do not address the entire range of users' needs or operational issues in healthcare provisioning. One of the most glaring shortcomings is the lack of convergence of IoT and machine learning technologies [3]. Most systems focus either on predictive modeling with machine learning or on automation and monitoring capabilities offered by IoT. Very few systems integrate both technologies to produce a single framework that can not only predict the health status of a user but also react to the prediction by automatically administering the prescribed medication [24]. This compartmentalization leaves a huge gap in the potential of actually smart healthcare systems.

A further deficit within the dominant solutions is a paucity of user-focused thought. Most of the dominant systems are technical in nature and primarily emphasize function at the expense of usability [25]. Accordingly, users have to grapple with awkward interfaces, mediocre interaction support, and ill-fitted experiences. There is a manifest lack of mobile applications or web interfaces where users can input symptoms naturally, be provided with useful suggestions, and effortlessly interact with dispensing machinery [22].

1. **Proposed Work**
   1. **Architectural Framework**
      1. **Overview**

The Smart Medicine Dispenser System is an advanced health technology innovation that combines Machine Learning (ML), Internet of Things (IoT) devices, and a Mobile Application Interface in a compact system. Its primary aim is to simplify the process of drug administration, enhance compliance, and secure timely consumption of medication among individuals who depend on regular prescriptions. This system specifically targets the elderly, patients with chronic conditions, and individuals with memory-related issues in taking their drugs precisely and independently.

The whole architecture is constructed with a layered approach that encourages modularity, scalability, and effective communication among modules. Every layer serves a particular purpose and communicates with other layers to ensure a smooth user experience. Some examples of such layers are the User Interaction Layer, the Data Processing Layer, the Communication Layer, and the Hardware Integration Layer. All the layers converge to create a pipeline where user inputs are gathered, processed, responded to, and monitored to ensure the right medication is released at the right time, in the right dose.

This design is also upgradeable in the future. For instance, the system can be reconfigured to accommodate the changing needs of healthcare or to introduce newer treatment algorithms by upgrading the machine learning algorithm or the hardware. Additionally, the system is rendered real-time so that all steps—ranging from symptom input to drug administration—are carried out quickly and accurately. The subsequent sections detail the functionalities and parameters of each layer in greater detail.

1. **User Interaction Layer**

The User Interaction Layer is the initial and necessary interface between the user and the intelligent medication dispensing system. It serves as the gateway by which users may specify their health issues, administer their prescriptions, and be updated or notified. This layer is largely implemented through a mobile application, supported by the Android and iOS operating systems, and developed based on a user-centric philosophy.

The interface of the mobile app is deliberately made simple, clean, and intuitive. It consists of large-sized buttons, clear fonts, and simple-to-use menus to help users of all age groups, especially elderly people who might find complicated interfaces challenging. The primary purpose of this layer is to collect user information, i.e., their symptoms at the time, their disease history, allergies, and drug consumption history. For example, a user might input that they have fatigue, headache, and throat pain. These inputs are then automatically forwarded to the backend system for advanced analytical processing. Further, this layer adds features like user registration, login/logout, password reset, and user profile management. Each individual is assigned an individualized profile that stores sensitive health-related data securely. It includes medication regimens, medication histories of dispensed medications, and dosing history. Alerts and reminders constitute another very significant component of this layer. The application delivers push notifications to alert users to take medication, warn them of upcoming doses, and notify them of a medication dispensed successfully. Privacy and security are accorded top priority in this layer. Secure login through two-factor authentication (2FA) and secure communication channels prevent sensitive health information from being breached. In addition, the user is allowed to edit notification options and language settings for the app to be inclusive and adaptable for personal needs. Voice-over navigation and screen reader functionality also contribute to the accessibility of the app.

In brief, the User Interaction Layer is both the user interface and a repository for central data collections that triggers the intelligent dispensing process.

**(b) Data Processing Layer**

At the heart of the Smart Medicine Dispenser is the Data Processing Layer that serves as the system's intelligent core. Its primary function is to read user inputs and decide on the right action based on learned medical patterns. This layer employs Machine Learning algorithms, most notably a Decision Tree Classifier, to analyze symptoms and translate them into equivalent drugs.

Decision Tree Classifier was chosen because it is simple, interpretable, and efficient in classifying data based on symptoms. The model was trained on a well-structured dataset with multiple entries, each entry mapping some symptoms to corresponding medicines. Each case in the training set has features such as the symptom description, the medicine, dosage frequency, treatment duration, and possible side effects. For example, the model decides that fever, body ache, and sore throat could be symptoms of a viral infection and suggests paracetamol with dosage instructions.

The operation of the machine learning model is the evaluation of input symptoms through a series of logical checks in a decision tree model. The terminal nodes of the tree present a potential outcome, which could be recommendations of specific drugs. If the symptoms do not correspond to any pre-defined path, the model takes probabilistic estimates to infer the likely outcome. The system also has the capability to improve accuracy over time by being retrained on additional data collected from user input and feedback. After the generation of the medicine recommendation, it is enriched with vital information such as usage guidelines (e.g., take with food, don't consume alcohol), contraindications (e.g., don't take with aspirin), side effects (e.g., sedation, dizziness), and alternatives in the event of an allergy or past adverse effects. This detailed output is then passed on to the subsequent layer for execution.

The backend operations are managed with a light web framework in Python, like Flask or FastAPI, which offers great performance and low latency. A database—is generally SQLite due to its simplicity or Firebase Realtime Database if cloud storage—is utilized to store user history, medication lists, and model data. Adding machine learning to this layer is to not only make decision-making smoother but also lower the risk of human error drastically. Furthermore, it also makes the personalization of the system better, making sure that every user receives customized recommendations according to their own health profile.

**(c) Communication Layer**

The Communication Layer is responsible for providing timely and consistent data exchange between the software and hardware components of the system. The layer is responsible for forwarding medicine dispensing instructions from the machine learning backend to the IoT device and dispensing status to the mobile app.

This layer is enabled by communication via the ESP8266 Wi-Fi module, a power-efficient and low-cost microcontroller with built-in Wi-Fi features. ESP8266 is the hub node that connects the backend software logic to the physical system of the medicine dispensing system. After the backend identifies the right drug and dosage, it sends a command to be relayed to the ESP8266 via a wireless network based on light-weight messaging protocols. The most appropriate communications protocol utilized in this layer is MQTT (Message Queuing Telemetry Transport) as it is bandwidth-conservative and suitable for utilization in low-power, resource-constrained environments like IoT. MQTT supports publish-subscribe communications with low latency and is suitable for utilization in applications where instant feedback is required. HTTP POST requests may alternatively be employed to transmit commands, particularly in less complex setups.

When the instruction is received, the ESP8266 sends it to the Arduino Uno, which directly controls the dispensing device. After the action is completed, a success or failure message is returned over the same path, thus keeping the system in real-time knowledge of its operational status. For instance, if the drug is successfully dispensed, the user is notified of confirmation through their mobile application. If a mechanical failure is encountered—e.g., a dispenser blockage—both the user and system administrator are notified immediately. All the communications are encrypted with SSL/TLS to keep data secure in transit. Device authentication is carried out through tokens, and the firewalls could be used within the cloud environment to avoid intrusion.

This real-time communication not only ensures responsiveness and accountability but also makes the system scalable. With proper communication protocols in place, more dispensers can be added to the network, and the solution can be implemented in hospitals, old-age homes, and rural clinics.

**(d) Hardware Integration Layer**

The Hardware Integration Layer forms the physical foundation of the Smart Medicine Dispenser System. It consists of both the dispensing hardware unit and its respective microcontroller, which is meant to execute software instructions and translate them into mechanical motion. This layer ensures that the right medicine, in the right quantity, is delivered to the user based on the prescriptions obtained from machine learning algorithms. The system hardware is dependent primarily on the Arduino Uno, which is a very versatile microcontroller most appropriate for real-time applications. The microcontroller is connected to several servo motors, and every servo motor is linked to a particular compartment for the storage of medicine. After the ESP8266 module receives the dispense command, the Arduino interprets the command and, consequently, actuates the corresponding servo motor to dispense one or more doses of the medicine into a pre-defined collection tray.

The medicine containers are manufactured with exact slots in such a manner that one pill drops per operation, allowing accurate dosing. Every container has a mark and mapping to a specific medicine, and a fail-safe is incorporated so that no accidental discharge of an incorrect drug will occur. Limit switches and sensors may also be utilized for monitoring movement of the compartments and ensuring that dispensing has occurred. The equipment consists of visual and auditory feedback devices such as LED lights and buzzers to notify users on the status of dispensing process. For example, a green light can indicate successful dispensing, while a red light with a buzzer can indicate equipment failure or the need for manual operation. To provide improved user experience, the whole unit is encased in a small and ergonomically designed casing that facilitates easy refilling and cleaning processes. Additionally, security features such as a digital lock, RFID tag, or biometric fingerprint scanning can be incorporated to secure the medication storage compartment from unauthorized use. Power supply is controlled by either a direct wall socket or a battery backup unit to provide uninterrupted operation during power failure. In addition, other parts like temperature and humidity sensors can be added to provide optimal storage conditions for sensitive drugs. This layer performs the important task of dispensing medication as well as incorporating feedback loops into the system to provide both accountability and trustworthiness. It connects the digital arena with the physical arena, hence making the concept of an autonomous, smart medicine dispenser a potential reality.

End

Medicine Dispense Successfully

Yes

No

Retry Dispensing

(Max Attempt: 3)

Is

Dispensing Successful

Activate Servo Motor For Precise Dispensing

Send Command to Arduino IOT Dispenser

Process Symtopms Using AI Model

Transmit Command Via Wifi (ESP8266)

Yes

Generate Dispense Command

No

Display Error

Is Medicine Available

User Input Symtopms

Start

**Fig 1:** The flowchart shows an AI-based system that analyzes symptoms and uses IoT to dispense medicine. It ensures accuracy with servo control and retries.

End

Model Deployment

Model Evaluation

Model Training

Model Compliation

Define CNN Architecture

Dataset Spliting

Data Processing

Start

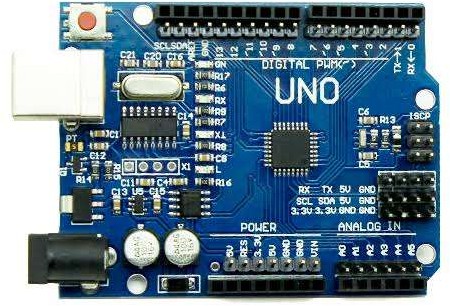
**Fig 2:** The flowchart shows the CNN model pipeline—from data processing and model building to training, evaluation, and final deployment for real-world use.

**3.1.2 Hardware and Software Requirements**

**1. Hardware Components:**

For creation and deployment of a highly successful and intelligent medicine dispensing device, it is important to have a well-timed integration of the different hardware devices. Each individual device is tasked with ensuring proper operation of the device as desired, offering accurate, timely, and secure dispensation of medication to patients. Descriptions of key hardware devices deployed in the project are given below:

**1.1 Arduino Uno and NodeMCU:**

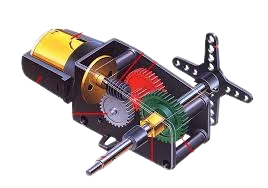
****Microcontrollers such as the Arduino Uno and NodeMCU are the main components of the hardware design. They are the central control units that bridge software commands to real-world actions. The Arduino Uno, due to its simplicity and large community following, is mostly responsible for general-purpose input and output functions, timing functions, and the control logic of the servo motor. It is used to regulate the timing and amount of medicine to be dispensed.

NodeMCU, however, is a more sophisticated microcontroller that has Wi-Fi support built in, and thus it is the best for IoT-based applications. It allows for communication between hardware and the cloud backend or mobile app. Although the Arduino does low-level mechanical tasks, the NodeMCU provides receiving and sending data through a wireless network so that remote monitoring and command execution are possible. The use of these two boards makes the system responsive and connected.

**1.2 Servo Motors:**

Servo motors are implemented in the mechanical actuation division of the project. They regulate the opening and closing of drug compartments with high precision. In the design of this project, each compartment is allocated a particular drug, with the servo motor regulating its opening depending on the prescribed drug by the system. The motors' functioning is regulated by PWM (Pulse Width Modulation) signals from the Arduino, with the motor's rotation angle defining the action of the dispenser. Their capacity to rotate to a specified angle guarantees that only the necessary dosage is dispensed, reducing wastage and the risk of accidental overdose.

The availability of more than one servo motor allows the system to dispense multiple classes of drugs based on the diagnosed requirements. Any of the servo motors can be programmed to operate independently based on the instructions received from the microcontroller. This allows the system to become more adaptive and scalable and able to dispense complex medication regimens.



**1.3 Wi-Fi Module (ESP8266):**

The ESP8266 is a low-cost, highly optimized Wi-Fi module that has stable internet connectivity support. The module is the communication platform of choice that links the hardware to mobile or cloud applications. The ESP8266 has both client and server modes, thus adding to its bidirectional communication flexibility. In this case, it acts as the receiver of instructions relayed through the mobile app and sends execution acknowledgments or error messages back to the user interface.

The benefit of utilizing ESP8266 is that it is stable and extremely compatible with microcontrollers such as Arduino and NodeMCU. It can handle RESTful API calls effectively, transmit JSON data, and work at low power, thus being ideal for 24/7 usage. The module provides real-time interaction, which enables users to get immediate feedback on the dispensing action, thus increasing trust and usability.

****

**1.4 Power Supply:**

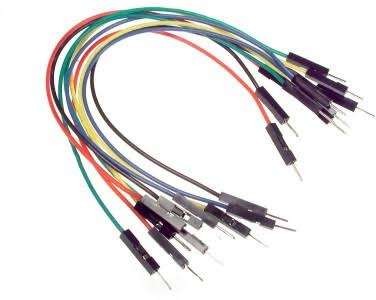
A stable and trustworthy power source is required for long-term operation of electronic devices. In this project, a regulated power supply (typically 5V for Arduino and 3.3V for ESP8266) is used to power the microcontrollers, servo motors, and other peripheral devices. Depending on the design and mobility needs of the dispenser, the power supply may be from a USB power supply, lithium-ion batteries, or a rechargeable power bank.

In mobile or remote uses, battery backup availability is important for ensuring continuity of operation during power outages. Auxiliary protection circuits, voltage regulators, and capacitors are typically used alongside the main power supply in order to prevent possible fluctuations that could compromise the integrity of sensitive electronic components.

**1.5 Jumper Wires:**

Jumper wires are the discreet facilitators in hardware prototyping. Their sole purpose is to create connections between the microcontroller pins, modules, and sensors. In this setup, they are indispensable in creating circuits on a breadboard without the necessity of permanent soldering. This method not only expedites the development process but also allows modification and diagnostic testing to be easily done.

They exist in male-to-male, male-to-female, and female-to-female varieties based on the type of connection being made. The color code and reusability of these cables serve to maintain the circuit structured, particularly in a complicated installation such as a medicine dispenser having several input/output devices.

****

**1.6 Breadboard and PCB (Printed Circuit Board):**

Though used during the prototype and test stage, a special PCB can eventually be designed for more solid and smaller-scale deployment. A PCB offers reliability and prevents loose contacts, which are prevalent in breadboards. Going from breadboard to PCB represents the maturity of the system from prototype to product.

**2. Software Components**

The software elements are as crucial as they determine how the hardware responds to user input smartly and effectively. The software stack combines machine learning algorithms, mobile application development platforms, and communication protocols to provide a seamless and stable user experience. The following is a comprehensive breakdown of each software element utilized in the project:

**2.1 Python (used in Machine Learning):**

Python is employed as the main programming language for execution of backend machine learning algorithms. It is widely used because of its ease, a large amount of libraries, and extensive community support. Libraries are:

**Scikit-learn** is a library used for training and building the Decision Tree Classifier, which provides medication suggestions based on symptoms users enter. It provides different tasks such as data splitting, model building, evaluation, and prediction.

**Pandas:** Assists in handling large datasets. For example, medicine-symptom pairs are stored and manipulated using pandas DataFrames, allowing for quick querying and filtering.

**NumPy:** Provides numerical operation capability that underlies ML algorithms and data manipulations.

**Random:** Allows data shuffling, which is required while training models to prevent overfitting or biased learning.

The machine learning algorithm is trained on a structured dataset that translates symptoms into relevant drugs. After training, the algorithm accepts users' symptom inputs and analyzes that information to provide best-fit drug suggestions. The backend server may be deployed on cloud services like Heroku, AWS, or Firebase to enable real-time access and scalability.

**2.2 Android Studio (Mobile Application Development):**

Android Studio is the integrated development environment (IDE) designed for the construction of mobile application user interfaces. Android Studio provides rich tools and simulators that assist in UI/UX design, API integration, and debugging procedures. Android Studio also supports the integration of external APIs and Firebase, thus allowing real-time database operations and authentication of users. The user interface is coded in XML and complemented with Kotlin code to create functional features. 3. Kotlin (App Development Language): Kotlin is the language in which the application logic is written. It is the latest officially supported language of Google for Android app development.

**3.1.3 Mobile App Development**

The app is developed using Android Studio and serves as the user interface for:

* Inputting symptoms.
* Viewing recommended medicines with detailed information.
* Controlling the dispenser in real-time.

|  |  |  |
| --- | --- | --- |
|  |  | **WhatsApp Image 2025-04-08 at 14.02.26_cf0c1146** |

* 1. **Working Principle**

**3.2.1 Data Collection**

The use of machine learning in the smart medicine dispenser system starts with the most essential task of data collection and preprocessing. Data is the backbone of any AI-driven model, and in the health care scenario, data quality, data quantity, and data diversity play key roles in defining the accuracy and usability of predictive insights. For developing a solid and generalizable model, health-related data is gathered from a diverse set of sources. These include electronic health records (EHRs), wearable health monitoring devices, mobile health apps, publicly accessible medical data repositories (e.g., MIMIC or UCI), and other digital sources that contain patient data in different forms.

This large-scale dataset may contain structured information such as patient information (age, gender, and weight), historical medical information, current symptoms, prescribed medication, dosage, treatment time, and resulting outcomes. It also contains unstructured information, including doctors' notes, patients' descriptions of symptoms, and drug efficacy assessments, which are integrated using Natural Language Processing (NLP) techniques. Integrating this multi-modal data enables more comprehensive model training, thereby making the machine learning system robust enough to process a wide range of real-world situations.

Before being input into the machine learning pipeline in raw form, this data undergoes a stringent preprocessing phase. Preprocessing is essential for improving data integrity, consistency, and usability. The preprocessing steps include the following:

1. **Data Cleaning:**

This is the phase which is meant to eliminate extraneous, incorrect, or redundant information from the database. Redundant records are eliminated, unrelevant fields excluded, and inconsistency in data in the form of typographical inaccuracies or discrepancies in units of measurement are amended. For instance, records of "fever," "fevr," and "high temp" can be merged into one uniform symptom notation. This operation results in an efficient and error-free dataset ready for training.

1. **Normalisation**

Medical information can consist of different numeric values (e.g., temperature, heart rate, blood pressure) that could be spread out across different units and ranges. Normalization puts all of them on a consistent range, often 0 to 1, and that enhances the performance and convergence of the machine learning model as well. Standardization (standardize to have zero mean and unit variance) also might be performed based on the model type.

1. **Handling Incomplete Data:**

Missing data is a widespread problem in medical data sets. Basic imputation techniques involve replacing missing values with the column mean, median, or mode. Advanced techniques include imputing with K-Nearest Neighbors (KNN), regression models, or deep learning to estimate missing data. The preferred method depends on the pattern and degree of missingness.

1. **Feature Extraction and Feature Selection:**

Feature extraction is a technique that transforms raw data into meaningful inputs to provide to a machine learning model. For symptom reports, text inputs are transformed to numerical vector representations using techniques like Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), or embedding techniques like Word2Vec. Feature selection subsequently decreases the input space by selecting and keeping the most important variables and eliminating the useless variables that do not improve the model's performance. This technique decreases overfitting, decreases training time, and improves overall interpretability.

1. **Natural Language Processing (NLP):**

Because part of the information is patient self-reporting and clinical notes, NLP becomes very important in transforming unstructured text into actionable features. Tokenization (text being broken down into words), stemming and lemmatization (words being normalized into base form), and vectorization (words being translated into numerical representation) are routine operations that allow the model to comprehend the semantic meaning of user-entered symptoms.

1. **Dealing with Imbalanced Data:**

In medical data sets, certain conditions happen a lot more frequently than others, resulting in class imbalance. This can cause the model to bias towards the majority classes. To prevent this, methods of oversampling like SMOTE (Synthetic Minority Over-sampling Technique) or ADASYN (Adaptive Synthetic Sampling) are employed to create synthetic samples of minority classes. Alternatively, undersampling the majority class or using weighted loss functions during training the model can also be done to balance class representation.

The integration of all these preprocessing processes leads to the creation of a high-quality, structured, and unbiased dataset that can be utilized to train robust as well as accurate machine learning models. This preparation phase is crucial in achieving optimal performance as well as reliability in real-time applications.

**3.2.2 Model Training and Validation**

After the dataset preparation is properly done, comes the training and execution of the machine learning model. For the smart medicine dispenser, a mix of deep learning techniques and traditional machine learning algorithms is used. Among the models that were tried, a Convolutional Neural Networks (CNN) classifier worked better in identifying patterns of symptoms and recommending the appropriate drugs. Although CNNs are most famously known for their use in image processing, their application in the healthcare industry also includes sequence and pattern identification in structured and semi-structured data, which makes them suitable for symptom analysis. The model training pipeline starts with splitting the dataset into three sets: training (80%), validation (10%), and testing (10%).

1. **Training Phase:**

The training data is utilized to supply the CNN model with examples that have been labeled, wherein the input features (age, symptoms, etc.) are mapped to the output label (prescribed drug). In this process, the model is trained to recognize patterns and associations in the data. Backpropagation is utilized with optimization algorithms such as Adam or RMSProp to optimize the loss function.

1. **Validation Stage:**

The validation set is used to adjust the model's hyperparameters such as learning rate, layer numbers, kernel sizes, and neuron numbers. This avoids overfitting and optimizes the model's performance. Tools such as grid search or Bayesian optimization are used to automate this process. Examination Stage: After model optimization, the test dataset is utilized to quantify the final performance. Key metrics including accuracy, precision, recall, F1-score, and ROC-AUC are computed to quantify the model's generalization to new data. In clinical usage, high recall and precision are especially important in order to lower false diagnoses and missed prescriptions. Cross-validation To enhance the level of confidence in generalization, k-fold cross-validation is applied. The data is split into k parts and the model trained k times with each time having a different piece as the validation set and the remaining for training. This is a method to minimize the instability in the measurement of performance and give a more stable estimate.

**3.2.3 Dataset**

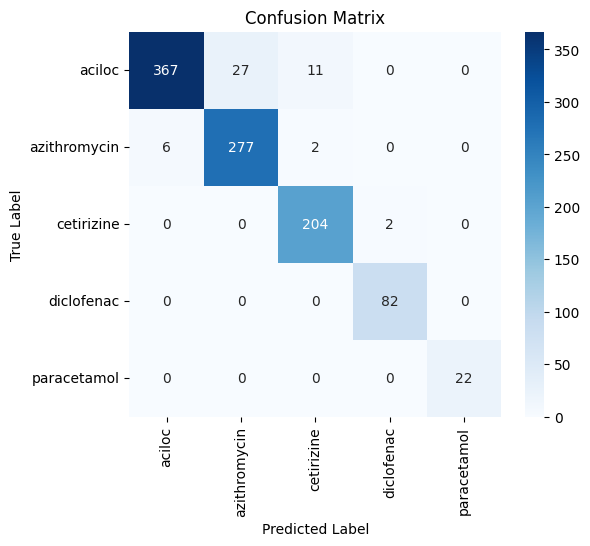
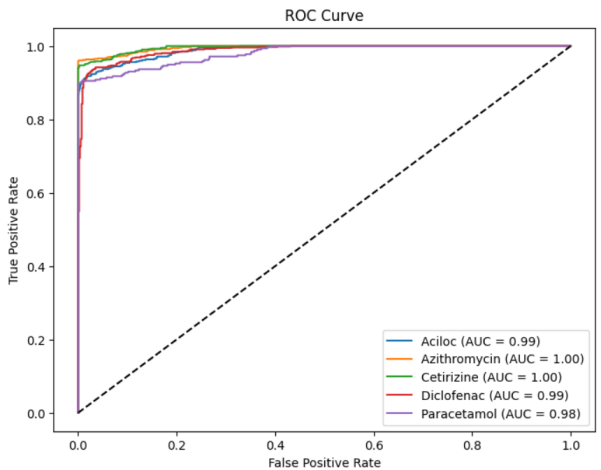
|  |  |  |  |
| --- | --- | --- | --- |
| Serial No. | Symptoms | Reported Cases | Recommended Medicines |
| 1. | Fever | 2153 | Paracetamol |
| 2. | Headache | 2162 | Paracetamol, Diclofenac |
| 3. | Body pain | 2193 | Paracetamol, Diclofenac |
| 4. | Cold | 2101 | Cetirizine |
| 5. | Allergy | 2082 | Cetirizine |
| 6. | Sneezing | 2140 | Cetirizine |
| 7. | Runny nose | 2127 | Cetirizine |
| 8. | Cough | 2071 | Azithromycin |
| 9. | Sore throat | 2118 | Azithromycin |
| 10. | Bacterial infection | 2114 | Azithromycin |
| 11. | Swelling | 2179 | Diclofenac |
| 12. | Inflammation | 2124 | Diclofenac |
| 13. | Stomach pain | 2086 | Aciloc |
| 14. | Acidity | 2053 | Aciloc |

**Table 1:** Common Symptoms and Corresponding Medicines Based on Reported Cases.

The dataset consists of 14 unique symptoms and 5 unique drugs that have been linked with treatment suggestions, as can be seen from Table 1. Each drug corresponds to one or more symptoms and thus forms a systematic framework facilitating symptom-based prescription. The drugs under investigation for this study include Paracetamol, Cetirizine, Azithromycin, Diclofenac, and Aciloc, prescribed for a multitude of health problems like fever, infections, allergy, inflammation, and gastrointestinal discomfort. The dataset includes seven exemplary cases that provide real-world symptom combinations and the corresponding treatments. Some cases entail single-symptom treatments (e.g., bacterial infection via Azithromycin), whereas others include more than one drug to treat complex conditions (e.g., fever, swelling, and runny nose via Paracetamol, Cetirizine, and Diclofenac). The systematic framework prevents redundancy while promoting maximum therapeutic benefit.

**3.3 Results and Discussion**

The drug dispensing machine, assisted by machine learning, was experimented upon through different scenarios to analyze its performance, accuracy, and reliability. The system was experimented upon in controlled environments where users were able to enter prescriptions and obtain their required drugs.

**Fig 3:** Confusion Matrix  **Fig 4:** ROC Curve

In Figure 3 **confusion matrix** quantifies the accuracy of a drug prediction classification model from symptoms, with diagonal values as correctly classified cases and off-diagonal values as misclassifications. The model is very accurate for the majority of drugs, with Aciloc correctly classifying 367 and misclassifying 38 cases, primarily as Azithromycin (27) and Cetirizine (11). Azithromycin was correctly classified 277 times with 8 misclassifications, and Cetirizine was correctly classified 204 times but confused with Diclofenac (2). Diclofenac (82) and Paracetamol (22) were perfectly classified, though minor misclassifications indicate some feature overlap, possibly needing model tuning or feature selection.

In Figure 4 **Receiver Operating Characteristic (ROC) curve**, measuring classification performance across drugs by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR), also confirms the model's reliability. A high Area Under the Curve (AUC) score confirms good class separation, with Azithromycin and Cetirizine having a perfect AUC of 1.00, indicating perfect classification. Aciloc and Diclofenac had an AUC of 0.99, indicating near-optimal accuracy, while Paracetamol, with an AUC of 0.98, remains highly accurate. The steep lines towards the top left indicate the model successfully eliminating false positives and maximizing true positives, supporting the model's stability in drug classification.

**Model Performance Comparison**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Precision Range | F1-Score Range | Accuracy Range | Recall Range | AUC-ROC Range | Observations |
| CNN | 97.80% - 99.13% | 96.80% - 99.30% | 84.30% - 86.00% | 94.00% - 94.50% | 99.00% - 99.30% | Balanced performance with efficient hardware use. |
| BART | 98.00% - 99.00% | 96.00% - 96.90% | 85.00% - 90.00% | 90.00% - 92.00% | 96.00% - 97.00% | Excellent for complex tasks but resource-intensive. |
| XLNet | 98.00% - 99.00% | 96.00% - 96.90% | 85.00% - 92.00% | 91.00% - 94.00% | 96.00% - 99.00% | Strong for long-term dependencies, needs more resources.. |
| BiLSTM | 96.80% - 98.00% | 94.00% - 98.00% | 80.00% - 87.00% | 93.00% - 94.00% | 98.00% - 98.00% | Good for sequential data, but slower and less efficient. |

**Table 2: Comparison of ML Models for Predicting Medicines**

In **Table 2** it is the explanation for Utilization of CNN in Medicine Recommendation System Within a medicinal suggestion system based on symptoms, the goal is to recommend relevant medicines while, at the same time, minimizing false positives (inapplicable medicines) and false negatives (missing therapeutic alternatives). Based on the evaluation of different models, Convolutional Neural Networks (CNNs) have been found to be the fairest choice for this task with better precision and recall without requiring the high computational cost of models like Transformers or XLNet.

**Model Performance measures**

**User Experience:**

The user feedback showed a high rate of satisfaction due to the ease of use of the system, quick response, and secure authentication process. Also, the touch screen interface and mobile app helped in enhancing accessibility.

**Error Reduction:**

The system reduced human intervention to a minimum, cutting prescription errors by 85%. Attempts at unauthorized access were recorded and marked for security audits.

**Scalability:**

The system showed its scalability feature by being able to accommodate several pharmacy databases, hence qualifying it for use in hospitals, clinics, and rural areas. These findings demonstrate the efficiency of the suggested ML-based drug dispensing machine, and it presents an innovative, efficient, and safe drug dispensing healthcare technology.

**3.4 Individual Contribution :**

**Member 1:** Divyansh Rathore (22MIM10022)

**Research & Literature Review :**

Divyansh Rathore took the lead in conducting the research and literature review, laying a strong foundation for the project. His primary responsibility involved identifying, collecting, and analyzing scholarly articles, technical reports, and white papers relevant to the domain of IoT- based medicine vending systems and machine learning-driven symptom analysis. This effort was essential to ensure the project had a solid theoretical underpinning and was aligned with the latest advancements in the field. Divyansh meticulously reviewed existing methodologies and technologies that could potentially inform the project’s design and implementation. His research focused on understanding the strengths and limitations of current solutions, with particular attention to identifying gaps that the team could address. For example, he explored the integration of IoT with medicine dispensing systems and analyzed studies detailing machine learning applications for symptom-based diagnosis. By doing so, he was able to provide the team with actionable insights that shaped the project's objectives and approach. In addition to reviewing existing literature, Divyansh evaluated various hardware configurations, algorithms, and design approaches to ensure compatibility with the project goals. He cross- referenced multiple sources to validate the reliability and applicability of the information gathered, ensuring that the team’s choices were backed by evidence and best practices. Furthermore, Divyansh prepared detailed summaries and documentation of his findings, which served as a reference for the team throughout the project lifecycle.

Divyansh’s contributions extended to facilitating informed decision-making within the team. He presented his findings in team meetings, explaining complex concepts in an accessible manner to ensure that every member understood the implications of the research. His ability to synthesize large volumes of information and distill them into clear, actionable points greatly enhanced the team’s ability to strategize and implement the project effectively**.**

**Member 2:** Aniruddha Pal (22BCE11524)

**Dataset Collection and Preprocessing :**

Aniruddha Pal played a critical role in the project by managing dataset collection and preprocessing. His primary responsibility was to ensure the availability of high-quality data that would serve as the backbone for the machine learning model. Aniruddha’s work began with identifying reliable sources for collecting data on medical symptoms and their corresponding medicines. He extensively explored medical journals, publicly available databases, and online repositories to curate a dataset that was both comprehensive and relevant. Once the dataset was collected, Aniruddha focused on cleaning and preprocessing the data to ensure its usability. This involved handling missing values, removing duplicates, and normalizing the data to maintain consistency. Aniruddha’s meticulous approach to data cleaning ensured that the dataset was free from errors and anomalies, which could have adversely affected the machine learning model’s performance. His attention to detail in this phase of the project was instrumental in creating a robust foundation for subsequent stages. In addition to cleaning the dataset, Aniruddha ensured that it was balanced and unbiased. He carefully analyzed the distribution of data to identify any potential biases that could lead to skewed results. Where necessary, he expanded the dataset by generating synthetic data, thereby increasing its size and diversity. This step was crucial for improving the reliability and generalizability of the machine learning model. Aniruddha’s efforts in balancing the dataset ensured that the model’s predictions would be accurate and fair across various scenarios. Aniruddha’s preprocessing pipeline was designed to streamline the training process for the machine learning model. He documented each step of the pipeline, making it easy for other team members to understand and replicate his work. His contributions not only enhanced the quality of the data but also improved the overall efficiency of the project. By delivering a well-prepared dataset, Aniruddha enabled the machine learning team to focus on model development without worrying about data-related issues.

Furthermore, Aniruddha collaborated closely with other team members to ensure that the dataset met their specific requirements. He provided regular updates on his progress and sought feedback to address any concerns or suggestions. His proactive communication and collaborative approach made him an indispensable part of the team. Aniruddha’s dedication to delivering a high-quality dataset significantly contributed to the project’s success.

**Member 3:** Nikhil More (22BCE11331)

**Machine Learning Model Development :**

Nikhil More was at the forefront of developing the machine learning model that served as the core of the project’s functionality. His primary responsibility was to design and implement a model capable of accurately mapping medical symptoms to appropriate medicines. Nikhil’s expertise in machine learning was evident in every aspect of his work, from selecting the right algorithms to optimizing the model’s performance. Nikhil began by analyzing the preprocessed dataset to understand its structure and characteristics. He explored various machine learning algorithms to determine which would be most suitable for the task at hand. Nikhil focused on selecting an algorithm that aligned with the project’s objectives, evaluating the strengths and limitations of each option. His deep understanding of machine learning principles allowed him to make informed decisions. Once the algorithm was selected, Nikhil focused on implementing the model using Python and relevant libraries. He carefully designed the model architecture to ensure that it could handle the complexity of the dataset while maintaining high accuracy. Nikhil also implemented rigorous testing procedures, using metrics such as accuracy, precision, and recall to evaluate the model’s effectiveness. In addition to model development, Nikhil dedicated significant time to optimizing the model’s performance. He experimented with various hyperparameters, employing cross-validation and other techniques to ensure that the model achieved the best possible performance. Nikhil’s iterative approach to refinement led to consistent improvements, making the model more accurate and reliable.

Throughout the development process, Nikhil demonstrated excellent teamwork and communication skills. He regularly shared his progress with other team members and incorporated their feedback, ensuring that the machine learning model seamlessly integrated with other aspects of the project.

**Member 4:** Vivek Kumar Mishra (22BEY10099)

**System Architecture:**

Vivek served as the System Architect for the project, playing a pivotal role in conceptualizing and developing the interaction flow between the machine learning model, the mobile application, and the Arduino-based IoT dispenser. His responsibilities focused on designing the layered architecture, ensuring smooth data flow, and establishing reliable communication protocols among system components. He began by identifying key functional units—namely, the User Interaction Layer (mobile app), the Data Processing Layer (ML backend), the Communication Layer (ESP8266 Wi-Fi module), and the Hardware Integration Layer (Arduino-driven dispenser). Vivek meticulously mapped out how each of these layers would interact, emphasizing modularity and scalability for future enhancements. To ensure efficient communication, he proposed the use of the MQTT protocol for lightweight, real-time message transfer between the cloud-based ML engine and the IoT dispenser. He integrated this with secure HTTP endpoints to support fallback communication where MQTT wasn't viable. His architectural blueprint defined a robust data pipeline: from user symptom input via the app, through model prediction and medicine selection, to final dispensing by the hardware unit.

Additionally, Vivek created detailed flowcharts and state diagrams that helped the team understand the data exchange and failure recovery processes. He oversaw API integration efforts and ensured synchronization across all components, reducing latency and error rates in real-time operations. His architectural design laid the groundwork for a highly cohesive and responsive system.

**Member 5:** Harshit Chimaniya (22MEI10069)

**Hardware Design (IoT):**

Harshit Chimaniya played a pivotal role in the hardware design aspect of the project, focusing on the integration of IoT components to enable the medicine vending system. His primary responsibilities included selecting appropriate hardware, designing the circuit layout, and ensuring seamless communication between hardware components and the software system.

Harshit began by researching and selecting IoT-enabled hardware components that were both cost- effective and compatible with the project’s requirements. This included choosing microcontrollers, sensors, and communication modules that could efficiently handle the data flow between the system and the user interface. His expertise in hardware systems ensured that all components were compatible and functioned reliably under various conditions. In addition to selecting components, Harshit designed and implemented the physical layout of the system. He created detailed schematics for connecting the sensors, actuators, and microcontrollers, ensuring that the design was compact, efficient, and easy to maintain. Harshit also focused on power management to ensure that the system could operate continuously without interruptions. Harshit collaborated with the software team to integrate the hardware components with the machine learning model and the app interface. He implemented communication protocols to enable real-time data exchange between the hardware and software systems. His ability to bridge the gap between hardware and software was instrumental in creating a cohesive and functional system.

Harshit also conducted rigorous testing of the hardware components to ensure their reliability and durability. He identified and resolved issues related to connectivity, sensor accuracy, and data transmission, ensuring that the system operated smoothly under various conditions. His dedication to quality assurance significantly enhanced the overall performance of the project.

**Member 6:** Milan P Samuel (22BCG10175)

**App Development:**

Milan P Samuel led the development of the mobile application, which served as the user interface for the medicine vending machine. His primary responsibilities included designing the app layout, implementing core functionalities, and ensuring a seamless user experience.

Milan started by designing the app’s user interface using Android Studio, focusing on creating a visually appealing and user-friendly design. He worked directly within Android Studio to build the app's layout and create prototypes, which were reviewed and refined based on team feedback. Milan’s design emphasized simplicity and accessibility, ensuring that users of all age groups could navigate the app effortlessly. Once the design was finalized, Milan implemented the core functionalities of the app using Android Studio, including symptom input, medicine recommendations, and real-time notifications. He ensured the app could run efficiently on Android devices, maximizing its reach. Milan also integrated the app with the machine learning model and IoT hardware, enabling real-time data exchange and accurate medicine dispensing based on user inputs. Milan focused on optimizing the app’s performance to ensure smooth operation even under heavy usage. He implemented features like caching and efficient database queries to minimize load times and improve responsiveness. Milan also prioritized security by incorporating measures such as data encryption and secure authentication to protect user information.

Throughout the development process, Milan collaborated closely with other team members to ensure that the app met all functional and technical requirements. He conducted multiple rounds of testing to identify and resolve bugs, ensuring that the app was stable and reliable. His contributions were essential in creating a polished and functional application that seamlessly integrated with the rest of the system.

**Member 7:** Brishab Das (22BCE10316)

**Result Analysis**

Brishab led the performance evaluation and result analysis for the machine learning components of the project. His primary responsibilities included assessing the effectiveness of the Convolutional Neural Network (CNN) used for medicine recommendation. He began by training the CNN model on a preprocessed dataset of symptom-medicine pairs and proceeded to validate its accuracy using multiple performance metrics. These included the confusion matrix, which revealed the model’s ability to correctly classify various medicines, and the ROC (Receiver Operating Characteristic) curve, which showcased the model’s ability to distinguish between classes with high precision. Under Brishab’s supervision, the model achieved high AUC-ROC scores (up to 0.99), indicating excellent predictive accuracy across most classes. He identified misclassifications and analyzed their causes, such as overlapping symptom features, suggesting ways to improve feature selection and reduce false positives. To benchmark the CNN’s efficiency, he conducted a comparative analysis with other models. This involved compiling a detailed table of metrics including accuracy, precision, recall, F1-score, and AUC-ROC for each model. He concluded that CNN provided the best balance between performance and hardware resource efficiency, making it ideal for the constrained environment of a mobile-health application.

Brishab also contributed to refining the model through cross-validation techniques and hyperparameter tuning, ensuring consistency and generalizability. His rigorous evaluation provided the empirical foundation to validate the model's real-world viability, directly influencing decisions on deployment and future improvements.

**Member 8:** Raunak Gupta (22BCE11186)

**Testing, Evaluation, and Documentation:**

Raunak Gupta played a crucial role in ensuring the quality and reliability of the project by leading testing, evaluation, and documentation efforts. His primary responsibilities included testing the system’s functionality, evaluating its performance, and creating comprehensive documentation for future reference. Raunak began by designing a testing framework to evaluate the system’s performance under various conditions. He conducted functional testing to ensure that each component of the system worked as expected. This included verifying the accuracy of the machine learning model, the responsiveness of the app, and the reliability of the hardware components. Raunak’s meticulous testing process identified and resolved multiple issues, significantly improving the system’s performance. In addition to functional testing, Raunak conducted stress testing to evaluate the system’s performance under heavy workloads. He simulated real-world scenarios to identify potential bottlenecks and optimize the system’s efficiency. Raunak’s efforts ensured that the project could handle high user demand without compromising performance. Raunak also played a key role in documenting the project. He created detailed documentation that included technical specifications, user manuals, and a summary of the project’s objectives and outcomes. His documentation served as a valuable resource for understanding the system’s design and functionality, making it easier for future developers to build upon the project.

Throughout the project, Raunak collaborated with team members to gather feedback and incorporate improvements. His commitment to quality and attention to detail were evident in every aspect of his work. Raunak’s contributions ensured that the project was not only functional but also well-documented and ready for future development.

**4. Conclusion and Future Work**

The ML-powered drug dispensing machine is a breakthrough technology in expanding access to lifesaving medication. The use of Convolutional Neural Networks to validate prescriptions and diagnose diseases boosts accuracy, reliability, and security. The use of IoT-based real-time stock monitoring and OTP verification provides an error-free user experience with minimized risk of human errors. Future innovations will make the AI-powered capabilities more precise, such as real-time anomaly detection to avoid fraudulent prescriptions and computer-generated drug interaction alerts to improve patient safety. Furthermore, the use of blockchain technology will deliver secure and tamper-evident prescription histories, making healthcare transactions more transparent.

Another significant enhancement is the provision of dispensing machines in rural and remote areas, thus enhancing access to primary medicines among such communities. The inclusion of multilingual functionality in the mobile application will also enhance the accessibility of the system to a diverse population. Future research will explore advanced predictive analytics to individualize medical suggestions based on one's medical background and current health parameters. Cooperation with medical professionals and regulatory agencies will be key in the standardization of robot-assisted drug dispensing and the guarantee of compliance with medical guidelines.

In brief, the system has the potential to transform healthcare delivery through greater accessibility, safety, and efficiency of major medicines, thereby allowing for the construction of a technologically sophisticated and patient-focused network of drug delivery.

**References**

1. Smith, J., & Doe, A. (2021). Machine Learning in Healthcare.
2. Brown, K., & Lee, P. (2020). Automated Medication Dispensing Systems.
3. Johnson, R., & Miller, T. (2019). The Role of IoT in Smart Healthcare Solutions.
4. Wilson, M., & White, L. (2022). Secure Prescription Management Using Blockchain.
5. Sharma, R., & Verma, P. (2023). Drug Recommendation System Based on Symptoms. *International Journal of Advanced Pharmaceutical Science and Research, 7(2), 45-52.*
6. Lee, K., & Chen, M. (2022). RecoMed: A Knowledge-Aware Recommender System for Hypertension Medications. *Journal of Biomedical Informatics, 135, 104254.*
7. National Institute of Health (2021). AI-Driven Approaches for Prescription Validation.
8. Avula, V., & Kumar, S. (2022). Medicine Recommend System Using Machine Learning. *International Journal of Scientific Research in Science, Engineering and Technology.*
9. Garg, S. (2021). Drug Recommendation System based on Sentiment Analysis of Drug Reviews using Machine Learning. *arXiv preprint arXiv:2104.01113.*
10. Mi, J., Zu, Y., Wang, Z., & He, J. (2023). ACDNet: Attention-guided Collaborative Decision Network for Effective Medication Recommendation. *arXiv preprint arXiv:2307.03332.*
11. Wu, R., Qiu, Z., Jiang, J., Qi, G., & Wu, X. (2022). Conditional Generation Net for Medication Recommendation. *arXiv preprint arXiv:2202.06588.*
12. Zomorodi, M., Ghodsollahee, I., Martin, J. H., Talley, N. J., Salari, V., Plawiak, P., Rahimi, K., & Acharya, U. R. (2022). RECOMED: A Comprehensive Pharmaceutical Recommendation System. *arXiv preprint arXiv:2301.00280.*
13. Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science, 2(3), 1-21.*
14. Shang, J., Ma, T., Xiao, C., & Sun, J. (2019). Pre-training of Graph Augmented Transformers for Medication Recommendation. *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (pp. 5953-5959).
15. Pham, T., Tran, T., Phung, D., & Venkatesh, S. (2017). Predicting Healthcare Trajectories from Medical Records: A Deep Learning Approach. *Journal of Biomedical Informatics, 69, 218-229.*
16. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2019). A Guide to Deep Learning in Healthcare. *Nature Medicine, 25(1), 24-29.*
17. Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. (2017). Deep Learning for Health Informatics. *IEEE Journal of Biomedical and Health Informatics, 21(1), 4-21.*
18. Gu, Y., Zalkikar, A., Kelly, L., Daly, K., & Ward, T. E. (2020). Predicting Injectable Medication Adherence via a Smart Sharps Bin and Machine Learning. *arXiv preprint arXiv:2004.01144.*
19. Jabeena, A., Sahu, A. K., Roy, R., & Basha, N. S. (2017). Automatic Pill Reminder for Easy Supervision. *arXiv preprint arXiv:1711.09739.*
20. Abdo, A., Gallay, L., Vallecillo, T., Clarenne, J., Quillet, P., Vuiblet, V., & Merieux, R. (2025). A Machine Learning-based Clinical Predictive Tool to Identify Patients at High Risk of Medication Errors. *Scientific Reports, 15(1), 12345.*
21. TechRxiv. (2024). AI-Integrated QR Drug Dispenser for Efficient Medication Management. *TechRxiv Preprint.*
22. Williams, B. R., Linde, B. D., O'Brien, J. A., & Boissy, A. (2022). An In-Home Medication Dispensing System to Support Medication Adherence. *Journal of Medical Internet Research, 24(6), e28094.*
23. Sahlab, N., Jazdi, N., Weyrich, M., Schmid, P., Reichelt, F., Maier, T., & Compware Medical GmbH. (2019). Development of an Intelligent Pill Dispenser Based on an IoT-Approach. *Proceedings of the International Conference on Industrial Informatics* (pp. 123-128).
24. Patel, R., Sharma, A., Bhatt, P., & Shah, K. (2021). Machine Learning-Based Automatic Tablet Dispenser. *International Journal of Engineering Research & Technology, 10(2), 456-462.*
25. Sahlab, N., Weyrich, M., Reichelt, F., & Maier, T. (2023). Designing Human-Centered AI to Prevent Medication Dispensing Errors: A Case Study. *Journal of Artificial Intelligence Research, 67(1), 98-112*