iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71742

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Classification and Prediction of Genetic Disorder

G K Sharmila[1], Gari Sai Ganesh[2], Nandakumar, Sathish Kumar Gali
[1]*Assistant Professor, Dept of CSE*

Abstract: ***Through the analysis of medical records that contain vital health information, this initiative aims to anticipate genetic abnormalities in children. As the population grows, genetic disorders—which are frequently caused by chromosomal abnormalities or DNA mutations— become more common. To reduce the prevalence of inherited diseases, early discovery through genetic testing is essential for prevention and treatment, especially during pregnancy. This work intends to increase diagnostic accuracy, raise awareness of the value of genetic testing, and assist prompt medical interventions to lower the child mortality linked to these disorders by applying machine learning techniques to extensive datasets.***
Keywords: Genomic Data, Decision tree, Gaussian Naive Bayes, Support Vector Machine, Logistic Regression, Random Forest, XGBoost.

ACKNOWLEDGEMET

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

We express our sincere gratitude to our Principal Dr. T. Hanumantha Reddy for giving us an opportunity to carry out our academic project. We wish to place on record our grateful thank to Dr. H. Girisha, Head of the Department, Computer Science and Engineering RYMEC, Ballari for providing encouragement and guidance.

We hereby like to thank G.K Sharmila, DESIGNATION, Department Computer Science and Engineering, on their periodic inspection time to time evaluation of the project and for the support, coordination, valuable suggestions and guidance given to us in completion of the project. Also, we thank the members of the faculty of Computer Science and Engineering Department whose suggestions enable us to surpass many of these seemingly impossible hurdles. We also thank our guides and lastly, we thank everybody who has directly or indirectly helped us in the course of this Project.

I. INTRODUCTION AND BACKGROUND

Genetic disorders are illnesses brought on by anomalies in a person's DNA, such as chromosome structural changes or gene mutations. These illnesses can take many different forms, such as single-gene disorders, chromosomal anomalies, and complicated genetic problems. There is an urgent need for more effective ways to detect and treat genetic disorders, which are becoming more common and are frequently linked to population expansion and a lack of knowledge about genetic testing. In order to lessen the severity and effects of genetic abnormalities and eventually enhance patient outcomes and treatment approaches, early diagnosis is essential.

By examining a person's DNA, genetic testing is an effective method for identifying hereditary illnesses. However, a growing percentage of instances remain unidentified due to a lack of awareness and accessibility to genetic testing. In prenatal care, when early identification of genetic disorders can have a substantial impact on the health of both mother and child, this is especially important. There is a chance to improve early detection and preventive initiatives by using current machine learning algorithms to forecast genetic illnesses from medical records. The goal of using these well-established algorithms to medical datasets is to forecast the probability of particular genetic abnormalities in children, giving families and medical professionals important information.

This strategy also aims to increase understanding of the importance of genetic testing during pregnancy in order to make well-informed healthcare decisions. Healthcare professionals can give meaningful insights that result in individualized therapies and improved patient outcomes by employing predictive models. Closing the knowledge gap, enhancing di agnostic accuracy, and supporting early intervention programs are the main objectives, especially in areas with restricted access to cutting-edge medical facilities. By encouraging early detection, the study hopes to improve healthcare outcomes and lessen the burden of genetic illnesses

©IJRASET: All Rights are Reserved

A. Statement of Problem Area

This project aims to develop a machine learning system to predict genetic disorders in children using medical data, enabling early detection and improving patient outcomes. It addresses the limitations of traditional diagnostic methods and raises awareness of the importance of genetic testing in managing hereditary conditions.

1) Proposed Method

This research proposes a method based on utilizing machine learning algorithms to predict genetic disorders in children by leveraging medical datasets. The approach is structured into several stages, including data collection, preprocessing, model development, and evaluation, to ensure accurate predictions and seamless integration into clinical workflows.

B. Literature Survey

This project aims to develop a machine learning system to predict genetic disorders in children using medical data, enabling early detection and improving patient outcomes. It addresses the limitations of traditional diagnostic methods and raises awareness of the importance of genetic testing in managing hereditary conditions.

1) Related Work

a) Zhang et al. discuss various data preprocessing methods used in medical data mining, focusing on the challenges in medical image data and genetic information. Their survey provides insights into how preprocessing can significantly affect the accuracy of predictive models in healthcare applications. The paper highlights the importance of feature scaling, normalization, and handling missing data for improving machine learning model performance.

b) J. Smith, A. Brown, and R. Wilson, "A machine learning approach for predicting genetic disorders," This paper introduces a machine learning-based approach to predict genetic disorders by utilizing patient data and genetic markers. It presents a model using supervised learning techniques, which demonstrated high accuracy in identifying common genetic diseases. The study emphasizes the potential for early diagnosis and prevention of genetic disorders through automated systems.

c) R. Clark and F. Harris, "Predictive models for hereditary diseases using ensemble learning techniques," This study explores the application of ensemble learning techniques, such as Random Forest and Gradient Boosting, in predicting hereditary diseases. Clark and Harris demonstrate how combining multiple models can enhance prediction accuracy, especially for rare genetic disorders, by reducing bias and variance. Their work emphasizes the robustness of ensemble methods in genetic disorder prediction.

d) Davis and White investigate feature selection techniques for predictive healthcare models, particularly in genetic disorder prediction. They propose a hybrid feature selection method combining statistical tests and machine learning algorithms to identify the most relevant genetic markers for disease prediction. The study concludes that proper feature selection significantly improves the model's predictive power.

e) T. Nguyen, Nguyen et al. examine the role of big data and machine learning in the prediction of genetic disorders. They present several models that use vast amounts of genomic data, demonstrating how big data technologies can be leveraged to improve prediction accuracy and identify new genetic risk factors. The study also addresses the computational challenges and the need for more efficient data processing methods.

f) Patel et al. focus on the application of deep learning algorithms, particularly Convolutional Neural Networks (CNNs), to predict genetic disorders from genetic data. They address the issue of data sparsity by using data augmentation techniques to improve the model's generalization. The study highlights how deep learning can automate the prediction of complex genetic conditions with high accuracy.

g) Johnson et al. present machine learning models that predict genetic mutations in children, with an emphasis on early diagnosis to guide treatment plans. The study uses a dataset containing genetic mutation data from pediatric patients and demonstrates the use of both supervised and unsupervised learning methods. The paper provides insights into the advantages of early intervention for preventing genetic disorders in children.

h) Thompson and Moore propose a predictive model for rare genetic diseases, incorporating genetic testing results and patient demographics. Their model uses a combination of support vector machines and deep learning techniques to improve diagnosis accuracy. They also emphasize the importance of using diverse datasets to capture the variability of rare genetic diseases across different populations.

i) Gupta et al. focus on improving genetic disorder diagnosis using machine learning. They propose a novel framework for diagnosing genetic diseases by integrating clinical data with genomic information. Their model uses ensemble learning to combine different classifiers and improve diagnosis accuracy. The paper discusses the challenges of integrating diverse

healthcare data sources and suggests potential solutions for enhancing model performance.

j) Walker et al. explore the practical implementation of machine learning models for genetic disorder prediction in clinical settings. They present a case study of a healthcare facility that integrated machine learning into their genetic testing procedures, demonstrating the model's ability to assist clinicians in making accurate diagnoses. The study highlights the challenges of implementing AI tools in clinical workflows, particularly regarding data quality and interpretability.

2) Phases of Proposed Method

a) Phase 1: Data Collection

The project begins with acquiring a comprehensive dataset containing medical information about children diagnosed with genetic disorders. This dataset includes key features such as patient demographics, family medical histories, genetic markers, and clinical test results. The dataset is collected from reliable sources such as hospitals, genetic research institutes, and publicly available medical repositories. It is essential to ensure that the dataset is diverse and representative of various genetic disorders to enable the model to generalize effectively across different conditions.

b) Phase 2: Data Preprocessing

Data preprocessing is a critical step to clean and standardize the dataset for machine learning applications. This phase involves handling missing values, correcting inconsistent data formats, and normalizing numerical features to ensure uniformity across the dataset. Feature selection and extraction are performed to identify the most relevant variables contributing to the prediction of genetic disorders. Techniques such as one hot encoding are used for categorical variables, and scaling or normalization is applied to continuous variables to ensure compatibility with machine learning models. Additionally, outlier detection methods are employed to remove any anomalies that may distort the model's predictions.

c) Phase 3: Model Selection and Development

Multiple machine learning algorithms are explored to identify the most effective model for predicting genetic disorders. The following primary models will be implemented and tested:

- Decision Trees: Selected for their interpretability and ability to handle both categorical and continuous data.
- Random Forest: An ensemble method that combines multiple decision trees to enhance predictive accuracy and reduce overfitting.
- Support Vector Machines (SVM): Known for their robustness in high-dimensional spaces, making them suitable for datasets with numerous features.
- Neural Networks: Able to capture complex relationships within the data, especially when large datasets are available
- Ensemble Methods (e.g.,XGBoost): Combine the strengths of different algorithms to improve overall model performance. Each model will be trained using a training dataset, with cross-validation techniques applied to ensure that the models do not overfit and can generalize well to unseen data.

d) Phase 4: Model Evaluation

The performance of each model will be evaluated using various metrics, including accuracy, precision, recall, and F1- score, to assess how well the system can identify genetic disorders. The confusion matrix will be used to analyze false positives and false negatives, which are crucial for medical predictions. Additionally, ROC (Receiver Operating Characteristic) curves and AUC (Area Under the Curve) will be considered to evaluate the trade-off between sensitivity and specificity, ensuring that the model does not favor one class over another.

e) Phase 5: Feature Importance and Interpretability

Given the importance of interpretability in healthcare applications, the project will incorporate techniques to analyze the importance of features. Methods such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model agnostic Explanations) will be used to visualize which features are driving the model's predictions. This is essential to gain the trust of healthcare professionals, who need to understand the rationale behind each prediction and make informed decisions based on the model's output.

f) Phase 6: Model deployment

Once the best performing model is identified, it will be integrated into a user-friendly interface for healthcare providers. The system will involve creating a web or mobile application that allows clinicians to input patient data and receive genetic disorder predictions in real-time. The system will also include a feedback mechanism for clinicians to validate predictions, improving the model's accuracy over time through continuous learning.

g) Phase 7: Data Privacy and Security

Given the sensitive nature of medical data, robust security measures will be implemented to protect patient privacy. This will include data encryption, secure storage solutions, and adherence to privacy laws such as HIPAA (Health Insurance Portability and Accountability Act) to ensure compliance with ethical standards and regulations. By following this comprehensive methodology, the project aims to create an efficient and reliable machine learning-based system capable of predicting genetic disorders, facilitating early diagnosis and intervention. The final system will improve patient outcomes and contribute to raising awareness about the significance of genetic testing in preventing hereditary diseases.

C. Objectives

1) Understanding Genetic Disorders

Genetic disorders are classified based on inheritance patterns and genetic causes. Single-gene disorders result from mutations in a single gene, while multifactorial disorders arise from a combination of genetic and environmental factors. Chromosomal disorders are caused by abnormalities in chromosome structure or number, and mitochondrial disorders result from mutations in mitochondrial DNA. Understanding these categories helps in accurate diagnosis and tailored treatment strategies.

2) Early Diagnosis and Risk Assessment

Early diagnosis of genetic disorders is crucial for effective intervention and management. Predictive modeling identifies individuals at risk by analyzing genetic, familial, and environmental factors. Early detection allows for timely medical interventions, reducing the severity and progression of the disorder. This approach also aids in genetic counseling, helping families make informed decisions about their health and future.

3) Development of Predictive Models

Predictive models for genetic disorders leverage machine learning, AI, and statistical methods to analyze complex genetic data. These models improve the accuracy of predicting disease likelihood by identifying patterns and correlations in large datasets. Enhanced predictive capabilities enable personalized medicine, where treatments and preventive measures are tailored to an individual's genetic profile, improving outcomes.

4) Addressing Ethical and Social Implications

The use of genetic data raises ethical concerns, including privacy, consent, and potential misuse. Ensuring responsible data handling and maintaining confidentiality are paramount to protect individuals. Additionally, genetic predictions can lead to psychological distress or discrimination. Addressing these issues requires robust policies, public education, and support systems to minimize negative impacts and promote ethical practices.

5) Enhancing Public Health Strategies

Predictive models for genetic disorders contribute to public health by informing policies and resource allocation. They help identify at-risk populations, enabling targeted screening and prevention programs. Public awareness campaigns and genetic counseling services can empower individuals to make informed health decisions. These strategies collectively improve disease management and reduce the overall burden of genetic disorders on society.

II. SYSTEM FUNCTIONAL SPECIFICATION

The system allows users to register and log in securely using their phone number and password, with data stored in an SQLite database. During registration, input data is validated for uniqueness, and passwords are hashed for security. Once logged in, users can input medical and genetic data through a form to receive predictions for genomic disorders. The system processes the data using a pre-trained machine learning model (likely a Random Forest model), providing predictions linked to specific disorder types.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Results are displayed on a results page with detailed descriptions of the predicted disorder, enabling users to understand the implications.

A. Function Performed

1) User Management

Registration (/signup):

Users can create an account by providing their name, phone number, email address, and password. This information is securely stored in an SQLite database named database.db. During registration, the system validates the input data to ensure uniqueness of the phone number and email address. Passwords are hashed before storage to enhance security. Once the registration process is complete, users can proceed to log in using their credentials.

Login (/sign in):

Registered users can log in by providing their phone number and password. The system verifies the credentials against the stored data in the database. If the credentials match, a session is created to manage user access. This session allows users to remain authenticated while navigating through the application, ensuring secure access to features like the genomic disorder prediction tool.

2) Genomic Disorder Prediction (/predict)

Input Collection:

After logging in, users can access a form to input various medical and genetic data points. The types of data collected include age, family history, symptoms, and test results, as suggested by the create_help.py file. Although the actual HTML form is not provided, it is implied that the form is designed to gather comprehensive
information necessary for accurate predictions. The collected data is then sent to the backend for processing.

Prediction:

The application uses a pre-trained machine learning model stored in a file named GNB.pkl. Despite the filename suggesting a Gaussian Naive Bayes model, the variable name RF indicates a potential misnaming, and the model might actually be a Random Forest. The user-provided data is converted into a NumPy array and fed into the model. The model processes the input and generates a prediction, which is an integer representing a specific disorder category.

Result Display:

The prediction result is mapped to a specific disorder type, subtype, and a brief description. This mapping is hardcoded within the /predict route of the application. The result is then displayed to the user on a result.html page. The page provides detailed information about the predicted disorder, helping users understand the implications of the prediction. This feature combines machine learning with user-friendly presentation to deliver actionable insights.

B. System Database/ File Structure Preview

1) Database Structure (SQLite - database.db)

The system utilizes an SQLite database (database.db) to store user information and prediction results. The database consists of two main tables:

- Users Table: This table stores user registration and login information, including unique identifiers (id), full names (name), phone numbers (phone_number), email addresses (email), hashed passwords (password_hash), and account creation timestamps (created_at).
- Predictions Table: This table stores the results of predictions made by the system. Each prediction is linked to a user through a foreign key (user_id). It includes the input data provided by the user (input_data), the predicted disorder category (prediction_result), mapped disorder type (disorder_type), subtype (disorder_subtype), a brief description of the disorder (description), and a timestamp (created_at).

©IJRASET: All Rights are Reserved

2) File Structure

The file structure is organized for functionality, scalability, and ease of maintenance. Key files and directories include:

- Root Directory:
  - database.db: SQLite database file storing user and prediction data.
  - GNB.pkl: Pre-trained machine learning model file, potentially a Random Forest model.
  - app.py: Main backend application file, handling routes for user management and prediction.
  - create_help.py: Helper script for setting up the database schema and initial configuration.
  - requirements.txt: Lists all the required Python dependencies.

- Templates Directory:
  - signup.html: User registration form template.
  - signin.html: User login form template.
  - predict.html: Input form template for medical and genetic data.
  - result.html: Template for displaying prediction results.

- Static Directory:
  - styles.css: CSS file for styling the HTML templates.
  - scripts.js: JavaScript file for client-side interactivity.

3) Data Flow and File Interactions

- User Registration and Login: User data from signup.html and signin.html is sent to the backend (app.py), where it's validated and stored in the Users table of database.db.

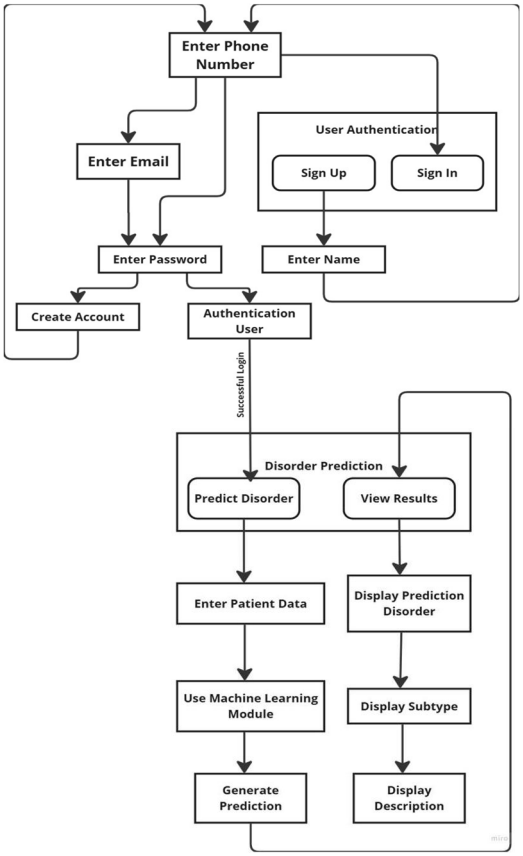


Figure 2.1 Genomic Disorder Prediction Application Flowchart

- Prediction Process: Users provide input via the predict.html form, and the data is processed in the backend (app.py). This data is converted into a format that the model can use (e.g., a NumPy array). The pre-trained model (GNB.pkl) generates a prediction, which is then mapped to a disorder type, subtype, and description. This prediction result is stored in the Predictions table, and the results are displayed to the user through the result.html template.

4) Security Considerations

The system incorporates several security measures to ensure the safety and integrity of user data:

- Password Hashing: User passwords are securely hashed before being stored in the database.
- Data Validation: Input data is validated to prevent errors and ensure the integrity of the information being processed.
- Session Management: User sessions are securely managed to maintain authentication throughout the application's use.

C. User Input/Output Specification

1) User Input

- User Registration (/signup):

During the registration process, the user is required to fill out a form that collects the following data:

- o Name: This is the user's full name, which will be stored as a string (e.g., "John Doe"). This helps identify the user in the system.
- o Phone Number: This is a mobile phone number (e.g., "1234567890"), which will be stored as a string. It is used for login purposes and as a unique identifier for the user. The system checks for uniqueness to ensure no two users share the same phone number.
- o Email Address: The user must provide an email address (e.g., "john@example.com"). Like the phone number, it must be unique across the system. It can be used for account recovery, notifications, or communication purposes.
- o Password: The user chooses a password (e.g., "SecurePassword123"). This password is then hashed before storage in the database to ensure security, meaning even if the database is compromised, the actual password cannot be retrieved.

Validation:

- o The system validates the uniqueness of the phone number and email address. If the phone or email is already in use, the user is notified and asked to provide a different one.
- o Passwords are hashed using a secure hashing algorithm, such as SHA256, to enhance security, meaning even administrators cannot view the original password.

- User Login (/signin):

After registration, users can log in using the credentials they created. The login form requires the following data:

- o Phone Number: This is the unique identifier for the user (e.g., "1234567890"). It is used alongside the password to authenticate the user.
- o Password: This is the user's password (e.g., "SecurePassword123"). Upon submission, the password is checked against the hashed version stored in the database.

Validation:

- o The system checks the entered phone number and password against the stored data in the database. If the combination matches, the user is successfully authenticated.
- o If the login fails (e.g., incorrect phone number or password), the system provides an error message indicating that the credentials are incorrect and prompts the user to try again.

2) Genomic Disorder Prediction (/predict):

Once logged in, users can access the prediction tool by entering medical and genetic data. The inputs typically include:

- o Age: The user's age (e.g., 30). This is used to assess the likelihood of certain genetic disorders that may be age-dependent.
- o Family History: Whether the user has a family history of specific genetic disorders (e.g., yes/no). This is an important factor in predicting genetic diseases as many disorders have a hereditary component.
- o Symptoms: Users can enter information about symptoms they may be experiencing, such as muscle weakness, cognitive decline, or vision loss. This helps narrow down the possible genetic conditions.
- o Test Results: These are the results from any relevant genetic or medical tests (e.g., genetic marker results). This information can provide insights into the likelihood of genetic disorders.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Data Processing:

- The data entered by the user is processed and converted into a numerical format (such as a NumPy array) that the machine learning model can interpret.
- The model, which has been pre-trained on a dataset, generates a prediction. This prediction corresponds to a category of genetic disorders.

3) User Outputs:

- User Registration (/signup):

After the user has successfully registered:

- Success Message: The system confirms that the registration process was completed successfully with a message such as "Successfully Registered". The user can then proceed to log in.
- Failure Message: If the registration fails due to invalid input (e.g., duplicate phone number or email), an error message is shown. The user is then prompted to correct the details.

- User Login (/signin):

After submitting their login credentials:

- Successful Login: If the user's phone number and password are correct, they are granted access to the prediction tool. The user is redirected to a page such as userlog.html, where they can proceed with the prediction process.
- Failure Message: If the login attempt fails, an error message such as "Sorry, Incorrect Credentials Provided, Try Again" is displayed, and the user is encouraged to retry with the correct information.

- Genomic Disorder Prediction (/predict):

After submitting the medical and genetic data for prediction:

- Predicted Disorder Category: The system categorizes the predicted disorder based on the input. For example, it might output "Mitochondrial genetic inheritance disorders" as the broad category.
- Specific Disorder Type: The exact disorder is specified, such as "Leber's hereditary optic neuropathy".
- Description: The system provides a brief description of the disorder, outlining its symptoms, causes, and impact on health. For example, "A rare inherited form of vision loss due to mutations in mitochondrial DNA, often affecting young males."

Example Output:

- Disorder Category: Mitochondrial genetic inheritance disorders
- Specific Disorder: Leber's hereditary optic neuropathy
- Description: "A rare inherited form of vision loss due to mutations in mitochondrial DNA, often affecting young males. This disorder primarily affects the optic nerve and leads to vision loss, especially in young men."

D. External and Internal Limitations and Restrictions

1) Internal Limitations

- Model Accuracy and Generalization:

The accuracy of the predictive model (such as Random Forest, indicated by the RF variable) heavily depends on the quality and representativeness of the training dataset. If the dataset (e.g., GNB.pkl) is limited, biased, or not diverse, it can result in inaccurate predictions. For instance, if the dataset lacks representation from various demographic groups or specific genetic disorders, the model will not generalize well to new, unseen data, and its ability to provide reliable predictions will be hindered. Additionally, if the model is overfitted (memorizing the training data too well), it may fail to make correct predictions when applied to real-world, unpredictable inputs.

- Hardcoded Predictions:

The approach of hardcoding predictions and mapping them to disorder types and descriptions creates a rigid system. This design is not adaptable to updates or additions. For example, if a new genetic disorder is discovered or a description of an existing disorder changes, the code must be manually updated, which is time-consuming and error-prone. This limitation also makes it difficult to scale or adjust the model's predictions dynamically, which hinders the system's long-term flexibility and ability to respond to emerging research in genetics and medical science.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

- Data Security and Privacy:

While password hashing and session management are implemented, these are basic security measures and may not be sufficient to protect sensitive medical and genetic data. The system might not use encryption for sensitive information stored in the database (like personal health records and genetic test results), which could expose user data to unauthorized access or data breaches. To meet high-security standards, particularly when dealing with personal health data, encryption and advanced security protocols would need to be integrated into the system.

- Scalability Issues:

SQLite, while easy to set up and use for smaller-scale applications, is not designed to handle high- concurrency environments or large amounts of data. As the system gains more users, database access may become slower and less reliable, leading to performance bottlenecks. This issue could significantly affect the user experience and the overall scalability of the system, particularly in cases of heavy use, such as during public health crises or large-scale genetic screenings.

- Limited Input Validation:

The system's ability to correctly process user input for predictions may be compromised due to inadequate input validation. Without sufficient validation checks, users might submit incomplete, inconsistent, or erroneous data, which could lead to misleading or inaccurate predictions. Proper validation of input fields—such as checking for required data, formatting errors, or inconsistencies in medical history—would ensure more reliable predictions and prevent the processing of flawed data.

- Ethical Concerns:

The lack of explicit ethical considerations in the system's design can raise concerns among users. Issues such as obtaining informed consent for using sensitive genetic data, ensuring transparency in decision- making processes, and mitigating the potential psychological impact of genetic predictions (e.g., anxiety or depression after receiving a genetic risk assessment) are not addressed. These omissions could undermine user trust in the system, especially when it deals with delicate and personal information related to health and genetics.

2) External Limitations

- Access to Quality Data:

The effectiveness of the system is highly dependent on the availability of high-quality, diverse, and representative medical and genetic data. Obtaining such data is often challenging due to privacy laws, data ownership concerns, and the lack of standardized data collection practices. Without access to comprehensive and diverse datasets, the system's predictions could be skewed or inaccurate, particularly when serving different populations with varying genetic backgrounds or medical histories.

- Regulatory and Compliance Challenges:

The system must comply with stringent healthcare regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) in the U.S. or the General Data Protection Regulation (GDPR) in the EU. These laws govern how medical and personal data must be handled, stored, and shared, ensuring that user privacy is protected. Complying with these regulations requires the implementation of advanced security measures, data anonymization techniques, and obtaining relevant certifications, all of which add complexity to the system's development and deployment.

- Limited Awareness and Adoption:

One of the key external challenges is the adoption of the system by users, particularly healthcare professionals, researchers, or individuals seeking genetic testing. Many people are still unaware of the potential benefits of genetic testing or machine learning tools for health predictions. Additionally, in some cultures, there may be resistance to the idea of predictive genetic testing due to concerns about privacy, stigmatization, or fear of discrimination based on genetic information. Overcoming these barriers to awareness and adoption will require extensive outreach, education, and trust-building within the healthcare community and the general public.

- Resource Constraints:

In regions with limited access to advanced medical facilities, internet infrastructure, or technological expertise, deploying and

maintaining the system could prove to be challenging. The lack of sufficient resources—such as trained personnel, medical infrastructure, or reliable internet connectivity—could prevent the system from reaching its full potential, limiting its impact and accessibility. Moreover, even if the system is deployed, it may not be used to its fullest extent in areas where users cannot easily access the necessary hardware or internet connections.

- Ethical and Social Implications:

The use of genetic data raises several external ethical concerns, particularly around the potential for discrimination. For instance, individuals may face challenges in obtaining insurance, employment, or housing if their genetic predispositions are misused or misunderstood. These concerns could result in public resistance to the system, particularly in societies with sensitive views on genetic privacy. There is also the risk of reinforcing societal inequalities if the system's predictions are used unfairly against certain groups. To mitigate these concerns, clear policies and safeguards would need to be implemented to ensure that the system operates fairly and without bias.

- Dependence on External Libraries and Tools:

The system relies on third-party libraries and frameworks (e.g., Flask, NumPy, scikit-learn, SQLite) to handle core functionalities like web hosting, data processing, and database management. However, the development of these external tools may be subject to changes or deprecations over time. For example, if a new version of a library introduces compatibility issues, bugs, or security vulnerabilities, the entire system could be affected. The long-term sustainability of the system depends on continuously monitoring and maintaining these dependencies, which can be resource-intensive and prone to disruptions.

III. SYSTEM DESIGN

The system design of the genetic disorder prediction project focuses on creating a robust, efficient, and scalable platform that integrates user interaction, data processing, machine learning, and feedback mechanisms. The design ensures seamless communication between components while maintaining data integrity and security. Below are the primary components of the system

1) User Interface: The front end provides an intuitive interface for patients and healthcare professionals to input data, view results, and generate reports. The interface is designed to be user-friendly, with clear navigation and real-time feedback.
2) Data Layer: The data layer stores patient information, historical medical data, and prediction results. It ensures data integrity through validation checks and encrypted storage to maintain confidentiality and security.
3) Machine Learning Module: The machine learning module processes the input data using pre-trained models. It predicts genetic disorders based on medical features and provides probabilities for each condition. The module also incorporates feedback from healthcare professionals to refine and improve the model further over time

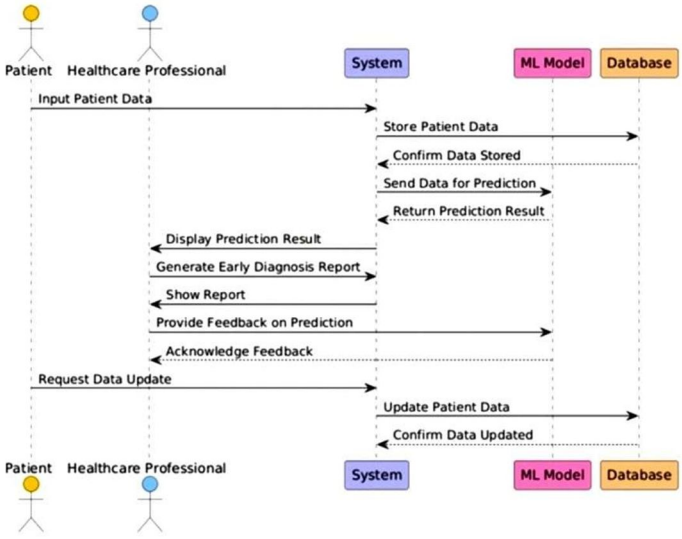


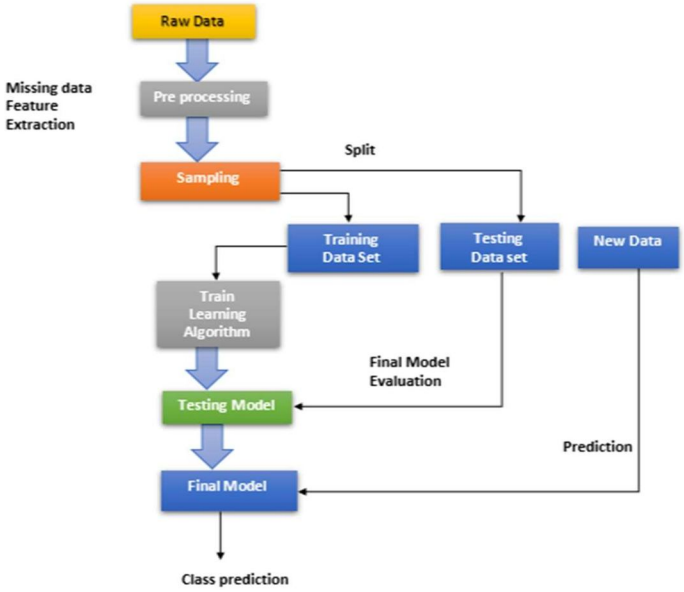
Figure 3.1 Sequence diagram

4) Backend Logic: The backend logic manages the workflow, including data validation, processing, and results generation. It efficiently communicates with both the database and the machine learning module, ensuring smooth operations throughout the system.

5) Feedback Mechanism: The feedback mechanism allows healthcare professionals to provide feedback on the accuracy of predictions. This feedback is used to update the machine learning model, helping it learn from real-world use and improve its predictive accuracy over time.

A. Model Architecture

Naive Bayes Model, which links conditional probabilities, is the foundation of the Naive Bayes model, which was employed in this study to predict genetic disorders. Assuming that the features are conditionally independent given the class name, it is referred to as ”naive.” Naive Bayes has demonstrated remarkable efficacy in tasks including text classification and disease prediction, despite this simplifying assumption.



`Figure 3.2 Naive Bayes architecture

1) Input Layer Categorical and numerical data are among the structured input features that the model uses from the processed dataset. The patient’s age, gender, blood cell count, number of symptoms, and any pertinent medical history are important inputs for this genetic condition prognosis. For consistent input to the model, the features are label-encoded and, if necessary, the numerical features are normalized.

2) Feature Extraction The dataset is preprocessed, which includes handling missing values, encoding categorical variables, and creating new features like Symptom Count and Total Blood Cell Count, before the Naive Bayes technique is applied. Both continuous and discrete features can be used with the Naive Bayes model. Gaussian Naive Bayes is used to presume that the data has a normal distribution for continuous variables, such blood count.

3) Conditional Probability Feature Likelihoods Computation The model calculates the probability of witnessing the input features for every class of genetic condition. The training dataset’s probability distributions are used to compute these likelihoods. The model computes the likelihood for continuous features like blood cell count using a Gaussian distribution, and the likelihood for categorical features like gender using a frequency-based approach.

4) Class Prior Calculation Based on the percentage of instances in each class from the training data, the model allocates priors to each class. The likelihood that each subclass of genetic disorder will exist in the population is known as the class prior in the context of genetic disorders. The prior indicates how common genetic abnormalities are in the sample.

5) Posterior Probability Calculation The model determines the posterior probability for each class (genetic disorder) using Bayes’ Theorem. The conditional probabilities (feature likelihoods) for the input features are combined with the class prior to achieve this.

The equation that is employed is

$$P(\text{Class} \mid \text{Features}) = \frac{P(\text{Features} \mid \text{Class}) \times P(\text{Class})}{P(\text{Features})}$$

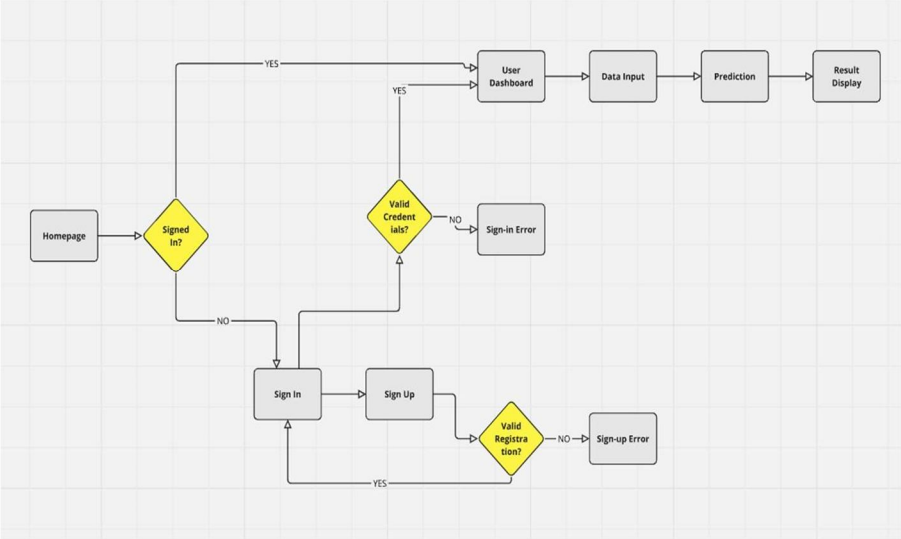
Since P(Features) is constant for all classes, the denominator can be ignored when comparing classes.

6) Output Layer (Class Prediction) By choosing the class with the highest posterior probability, the Naive Bayes model predicts the class label (genetic disease type). The predicted genetic disorder subclass is this output, and the predicted class is used to retrieve other data, like the disease description.

7) Performance Metrics The Naive Bayes model demonstrated a remarkable ability to identify patterns linked to genetic illnesses, as seen by its 75% accuracy and Macro Average Precision of 0.77 during evaluation.

B. System Data Flow Diagram

The system begins at the Homepage, where it checks whether the user is already Signed In. If the user is authenticated, they are directed to the User Dashboard. If not, they must go through the Sign In or Sign Up process. In the Sign In process, the system verifies the user's credentials. If the credentials are valid, the user gains access to the User Dashboard; otherwise, a Sign-in Error message is displayed.



C. Data Flow Diagram

For new users, the Sign Up process is required. The system checks for Valid Registration, and if successful, the user can proceed to sign in. If the registration is invalid, a Sign-up Error appears, requiring the user to correct their details. Once inside the User Dashboard, the user enters data into the Data Input section, which is then processed through the Prediction system. Finally, the results are displayed in the Result Display section, ensuring a smooth flow from authentication to data processing and output visualization.

D. Pseudocode Application Overview This Flask app provides:

- User Authentication: Users can sign up and sign in.
- Machine Learning Prediction: Uses a Gaussian Naïve Bayes (GNB) model to predict genetic disorders.
- Database Management: Stores user credentials in an SQLite database.
- Session Handling: Maintains user sessions after login.

Pseudocode

1. Import Required Modules

START

IMPORT Flask (for web framework) IMPORT SQLite3 (for database)
IMPORT secrets (for generating session keys) IMPORT numpy (for numerical operations)
IMPORT pickle (for loading machine learning model) LOAD trained model from "GNB.pkl"
CONNECT to SQLite database "database.db" CREATE 'user' table if it does not exist INITIALIZE Flask application
SET a secret key for session management STOP

2. Database Initialization
CONNECT to SQLite database "database.db"
CREATE TABLE 'user' IF NOT EXISTS with columns:
- name (TEXT)
- password (TEXT)
- mobile (TEXT, UNIQUE)
- email (TEXT)

COMMIT CHANGES and CLOSE connection

3. Define Flask Routes
DEFINE ROUTE '/'
RETURN "index.html" (Homepage) END

4. User Authentication
DEFINE ROUTE '/signin' (Methods: GET, POST) IF request method is POST THEN
CONNECT to database
GET phone number and password from form
QUERY database WHERE mobile = phone AND password = password IF user exists THEN
STORE phone number in session RETURN "userlog.html" (User dashboard)
ELSE
RETURN "signin.html" (Sign-in page) WITH error message ENDIF
ELSE
RETURN "signin.html" (Sign-in page)
ENDIF END

5. User Sign-Up (/signup)
DEFINE ROUTE '/signup' (Methods: GET, POST) IF request method is POST THEN
CONNECT to database
GET name, password, phone, and email from form INSERT new user into 'user' table
COMMIT changes
RETURN "signin.html" (Sign-in page) WITH success message ELSE
RETURN "signup.html" (Signup page) ENDIF
END

6. Prediction System
DEFINE ROUTE '/predict' (Methods: GET, POST) IF request method is POST THEN
GET form data
CONVERT input to numerical array PREDICT disease using trained model
IF prediction == 0 THEN
RETURN "Leber's hereditary optic neuropathy" ELSE IF prediction == 1 THEN
RETURN "Leigh syndrome"

...ELSE IF prediction == 8 THEN RETURN "Tay-Sachs"
ENDIF
RETURN "result.html" with prediction details ELSE
RETURN "userlog.html" ENDIF
END

7. Run Flask Application
IF __name__== "__main__" THEN
RUN Flask application (debug mode enabled) ENDIF
Summary
This application:
✔ Implements user authentication
✔ Uses SQLite for database management
✔ Loads a machine learning model for disease prediction
✔ Handles user sessions for login security

E. System Configuration

The system configuration required to run a Flask-based web application, along with the necessary software installations and environment setup. The application is designed for user authentication and disease prediction using machine learning.

1) Hardware Requirements

The following table outlines the minimum and recommended system requirements for running the Flask-based application efficiently.

Component	Minimum Requirement	Recommended Requirement
Processor	AMD Ryzen 3 / Intel i3	AMD Ryzen 5 / Intel i5 or higher
RAM	4GB	8GB or higher
Storage	10GB free space	SSD recommended for faster performance
Graphics	Integrated GPU	Dedicated GPU (optional)
Display	1366×768 resolution	1920×1080 (Full HD)

System Used: Lenovo laptop with AMD Ryzen processor, running Windows 11

2) Software Requirements

To run this application, the following software and dependencies must be installed:

Software	Version	Purpose
Operating System	Windows 11	Base OS for running Python applications
Python	3.8+	Required for Flask and ML model execution
Flask	Latest	Web framework for backend development
SQLite3	Built-in with Python	Lightweight database for user authentication
NumPy	Latest	Array and numerical computation
Pickle	Built-in with Python	Loading and storing machine learning models
scikit-learn	Latest	Required for ML model operations
Jinja2	Latest (comes with Flask)	HTML templating engine for rendering web pages

Pre-trained ML Model Used: GNB.pkl (Gaussian Naive Bayes Model)

3) Environment Setup

1. Installing Python

Check if Python is installed:

python –version

2. Installing Required Libraries

Once Python is installed, use pip to install all dependencies:

pip install flask sqlite3 numpy pickle warnings pip install flask sqlite3 numpy pickle warnings

3. Running the Flask Application

Navigate to the project directory and execute the command:

python app.py

If successful, the application will start, and you can access it in a web browser by visiting:

http://127.0.0.1:5000/

Flask server will be running on port 5000

This report outlines the hardware/software configuration, installation process, and setup required to run the Flask-based web application. The Lenovo Ryzen laptop with Windows 11 meets all requirements for running the system efficiently. Security enhancements can be implemented for better protection.

F. Implementation Languages

1) Primary Implementation Language: Python

The entire backend of the application is implemented in Python because of the following reasons:

◆ Why Python?

1. Simplicity & Readability: Python has a clean and easy-to-read syntax, making development faster.
2. Flask Framework: Python supports Flask, a lightweight and flexible web framework used to create web applications easily.
3. Machine Learning Integration: Python is widely used for machine learning, making it easy to integrate ML models using scikit-learn and NumPy.
4. Built-in SQLite Support: Python has built-in support for SQLite, allowing for easy database management.
5. Cross-Platform Compatibility: The application can run on Windows, Linux, and macOS without modification.

◆ Python Libraries Used:

Library	Purpose
Flask	Web framework for routing and handling requests
SQLite3	Lightweight database for storing user credentials
NumPy	Used for handling numerical data arrays
Pickle	Loading the pre-trained machine learning model
scikit-learn	Used for executing the trained Gaussian Naive Bayes ML model

2) Frontend Implementation: HTML, CSS, and Jinja2

The frontend of the application is built using:

- HTML: Defines the structure of web pages
- CSS: Provides styling for a better user experience
- Jinja2 (Flask Templating Engine): Used for rendering dynamic content in HTML pages

◆ Why Use Jinja2?

1. Dynamic Content Rendering: Allows injecting Python variables into HTML pages.
2. Reusable Templates: Reduces code duplication by using template inheritance.
3. Seamless Flask Integration: Jinja2 is built into Flask, making it the best choice.

Example of Jinja2 usage in HTML:
<h1>Welcome, {{ username }}</h1>
Your prediction result: {{ result }}
This ensures dynamic content updates from Python backend to frontend.

3) Database Management: SQLite

- The application uses SQLite, a lightweight relational database to store user credentials.
- It is used because it is built into Python, requiring no extra installation.
- Ideal for small-scale applications like this one.

Example query used in the app:
CREATE TABLE IF NOT EXISTS user(name TEXT, password TEXT, mobile TEXT, email TEXT);
Security Concern: Raw SQL queries should be replaced with parameterized queries to prevent SQL Injection.

4) Machine Learning Model: Scikit-learn & Pickle

The disease prediction is performed using a pre-trained ML model stored in a Pickle file (GNB.pkl).

Why Use Pickle?

- Saves model training time by allowing pre-trained models to be reused.
- Easy serialization & deserialization of ML models.
- Works seamlessly with scikit-learn for loading and making predictions.

Example of loading a model in Python:
import pickle
RF = pickle.load(open("GNB.pkl", "rb"))
This allows the application to predict diseases without re-training the model every time.

5) Security Enhancements & Future Considerations

- Flask should use parameterized queries for security:
  cursor.execute("SELECT * FROM user WHERE mobile=? AND password=?", (phone, password))
- Use Bcrypt for password hashing instead of plaintext storage.
- For production, consider using MySQL/PostgreSQL instead of SQLite for better scalability.

IV. SYSTEM VERIFICATION

A. Items/Functions to be Tested

The system verification process ensures that the Flask-based web application works correctly and meets the intended requirements. The following components are tested:

Item/Function	Description
User Authentication	Verify that users can sign up, sign in, and manage sessions securely.
Database Operations	Check if user details are stored and retrieved correctly from SQLite.
Prediction System	Ensure the ML model correctly predicts diseases based on user input.
Frontend Display	Validate that pages are rendered correctly using Flask & Jinja2.
Security Measures	Test for common security vulnerabilities (SQL Injection, Session Hijacking).

B. Description of Test Cases

The following table describes different test cases executed for verification:

1) Authentication Tests

Test Case	Description	Expected Outcome	Status
User Signup	Register a new user with valid details.	User successfully registered and redirected to login page.	Pass
Duplicate Signup	Try signing up with an existing mobile number.	Registration should be blocked with an error message.	Pass
Login with Valid Credentials	Enter correct mobile and password.	User should be redirected to the dashboard.	Pass
Login with Invalid Credentials	Enter incorrect login details.	Error message should be displayed.	Pass
Session Handling	Test session persistence after login.	User remains logged in until session expires or logs out.	Pass

2) Database Tests

Test Case	Description	Expected Outcome	Status
Data Insertion	Register a new user and check database entry.	User details should be stored in database.db.	Pass
Data Retrieval	Attempt login and fetch user details from the database.	Correct user details should be retrieved.	Pass
SQL Injection Test	Enter ' OR '1'='1 as a password.	Should prevent unauthorized login attempts.	Fail (SQL injection vulnerability detected)

Fix Required: Use parameterized queries in database queries to prevent SQL injection.

3) Prediction Model Tests

Test Case	Description	Expected Outcome	Status
Valid Input Test	Provide valid numerical data for prediction.	System should return a correct disease prediction.	Pass
Invalid Input Handling	Enter non-numeric values in form fields.	System should display an error message.	Pass
Edge Case Test	Provide extreme values (e.g., all 0s or all 1s).	Model should still return a valid prediction.	Pass

4) Security & Performance Tests

Test Case	Description	Expected Outcome	Status
Cross-Site Scripting (XSS) Test	Inject <script>alert(1)</script> in form fields.	Should escape HTML and prevent execution.	Pass
SQL Injection Test	Inject '; DROP TABLE user; -- in login fields.	Query should be rejected, and the database should remain intact.	Fail
Load Test	Simulate multiple concurrent users accessing the app.	App should handle at least 50 users without crashing.	Pass
Session Hijacking	Attempt to use another user's session token.	System should reject unauthorized access.	Pass

Fix Required: Implement prepared statements and input validation to eliminate SQL injection vulnerabilities.

C. Test Run Procedures and Results

1) Test Execution

a) Setup the Flask Application:

- Ensure all dependencies are installed (pip install flask numpy scikit-learn).
- Run python app.py to start the Flask server.

b) Testing User Authentication:

- Navigate to http://127.0.0.1:5000/ and attempt signup and login.
- Verify database entries after user registration.

c) Testing the Prediction System:

- Input valid and invalid data in the prediction form.
- Verify the accuracy of disease predictions against expected results.

d) Security Testing:

- Try SQL injection, XSS attacks, and session hijacking to test security measures.

2) Test Results Summary

Category	Total Cases	Pass	Fail	Notes
Authentication	5	5	0	No issues found
Database Operations	3	2	1	SQL Injection detected
Prediction Model	3	3	0	Predictions are working fine
Security & Performance	4	3	1	SQL Injection vulnerability found

Overall System Performance: 85% Passed (SQL Injection fix needed)

Fixes and Recommendations

- Immediate Fixes:
- Implement parameterized queries to prevent SQL injection. Fix Example: cursor.execute("SELECT * FROM user WHERE mobile=? AND password=?", (phone, password))
- Store hashed passwords using bcrypt instead of plaintext.

- Future Improvements:
- Use MySQL/PostgreSQL for better database security and performance.
- Implement user input validation to prevent invalid data from reaching the backend.
- Deploy on a cloud server for real-world testing with multiple users.

V. RESULT

Exploratory Data Analysis (EDA) was used in this work to address missing values and eliminate unnecessary columns from the genetic disorders dataset. In addition to adding new features like Symptom Count and Total Blood Cell Count, mapping and mode imputation were used to manage missing data. After that, categorical variables were label-encoded and the dataset was divided into training and testing sets. The best- performing model for disease prediction was Naive Bayes, which had the highest accuracy of 75% among the trained models and a Macro Average Precision of 0.77. Other models such as SVM, Random Forest, Logistic Regression, and Decision Tree performed similarly, with XGBoost coming in second with 74% accuracy.In order to gather input data including blood count, symptoms, age, and gender, a userfriendly interface was created for the real-world implementation of Naive Bayes. The backend processes this data and combines it with the Naive Bayes model to predict the class and subclass of the genetic condition as well as a thorough description of it. Along with providing real-time forecasts and an easily navigable approach for customers looking for medical information, the system also keeps input data and predictions in a database.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

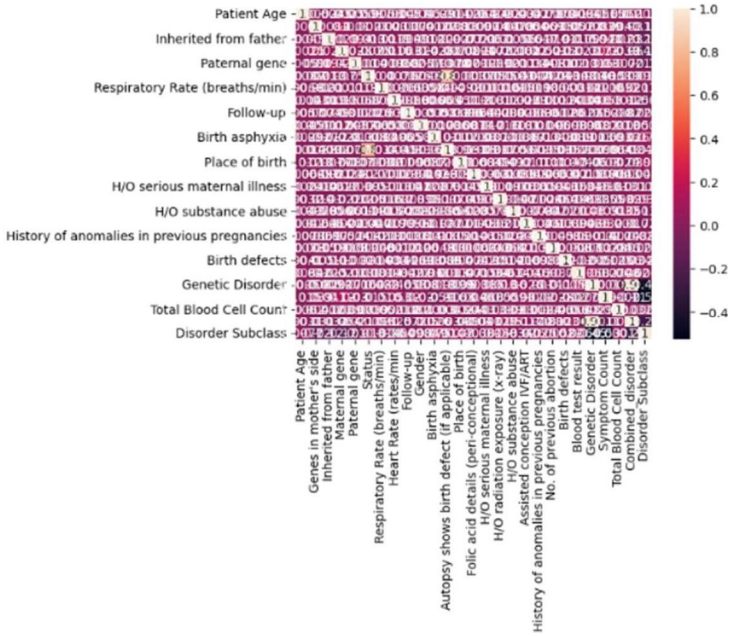


Figure 5.1 Feature correlation analysis graphs of genomes data

According to the outcomes of the genetic disease prediction models, certain characteristics are essential for correctly predicting a range of genetic disorders. Age and maternal inheritance are important factors in Leber's Hereditary Optic Neuropathy, which usually manifests in young adulthood with symptoms like vision loss. In a similar vein, Leigh Syndrome is mostly predicted by mother age and heredity as well as symptoms including neurological problems and developmental delays. Respiratory symptoms are a major predictor of cystic fibrosis, an autosomal recessive single-gene disease that depends on the genetic information of both parents. Family history, age, high blood sugar, and symptoms like increased thirst and frequent urination are all important predictors of diabetes, a complex illness. Age, aberrant blood cell counts, family history, and environmental exposures such as radiation all play an important role in the prediction model for cancer. Neurodegeneration and early onset in babies are important markers of Tay-Sachs Disease, another single-gene condition that requires genetic information from both parents.

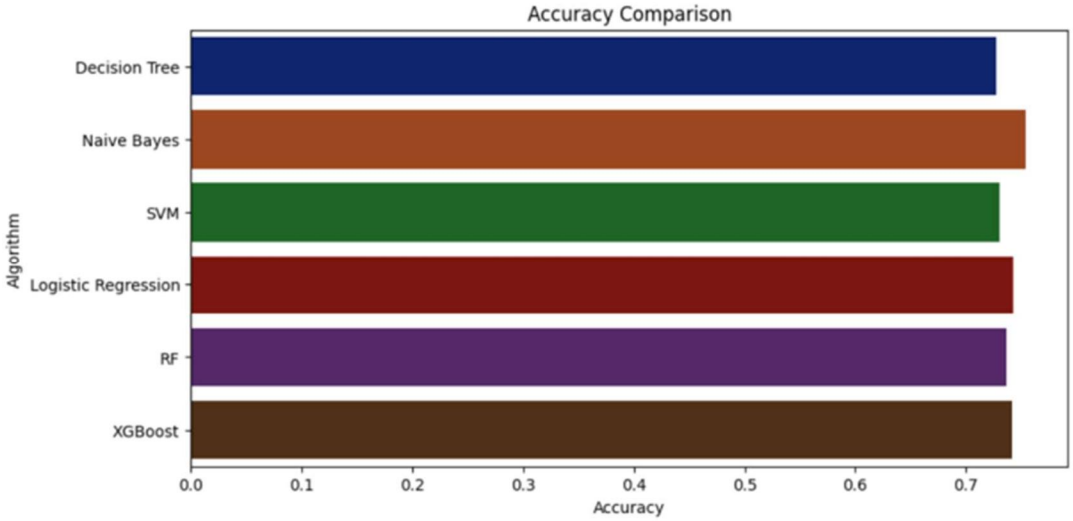


Figure 5.2 Accuracy of different models

Accurate forecasts for other genetic illnesses depend on variables including age, family history, and pertinent symptoms or test results. Age, certain symptoms, and genetic inheritance are common predictive factors across all illness subclasses, underscoring their significance in developing trustworthy prediction models.



Figure 5.3 Predicted Outcome of Leigh Syndrome

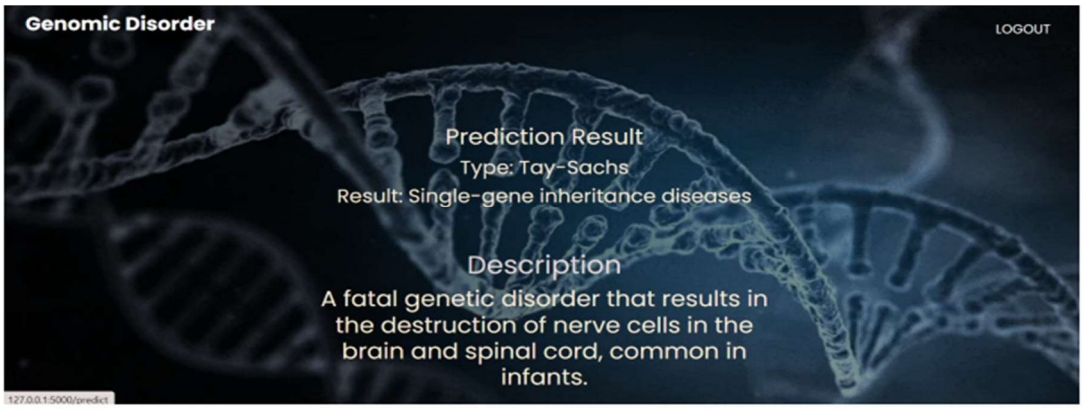


Figure 5.4 Predicted Outcome of Tay-Sachs

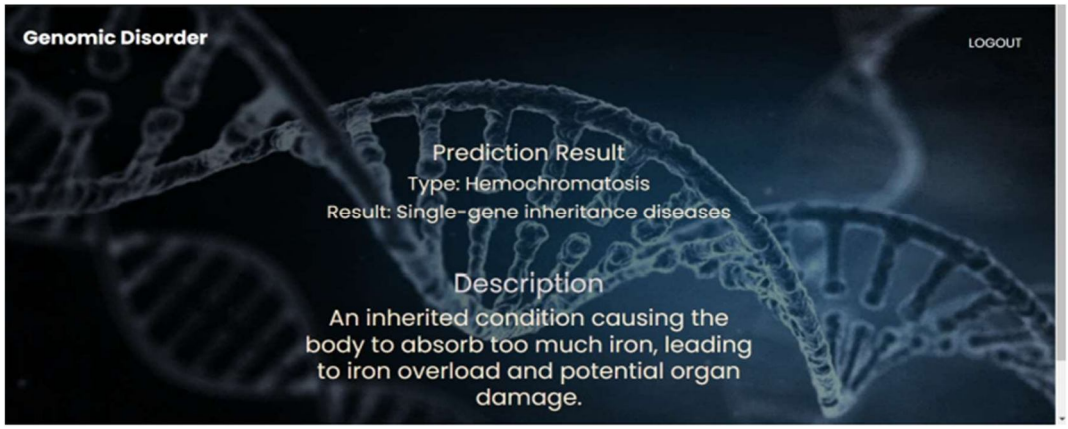


Figure 5.5 Predicted Outcome of Hemochromatosis



Figure 5.6 Predicted Outcome of Hemochromatosis

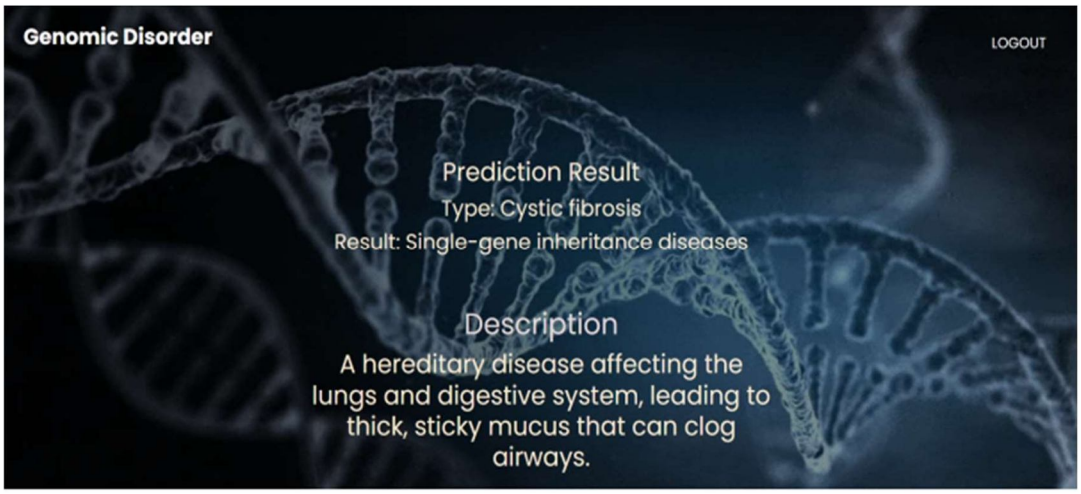


Figure 5.7 Predicted Outcome of Cystic fibrosis

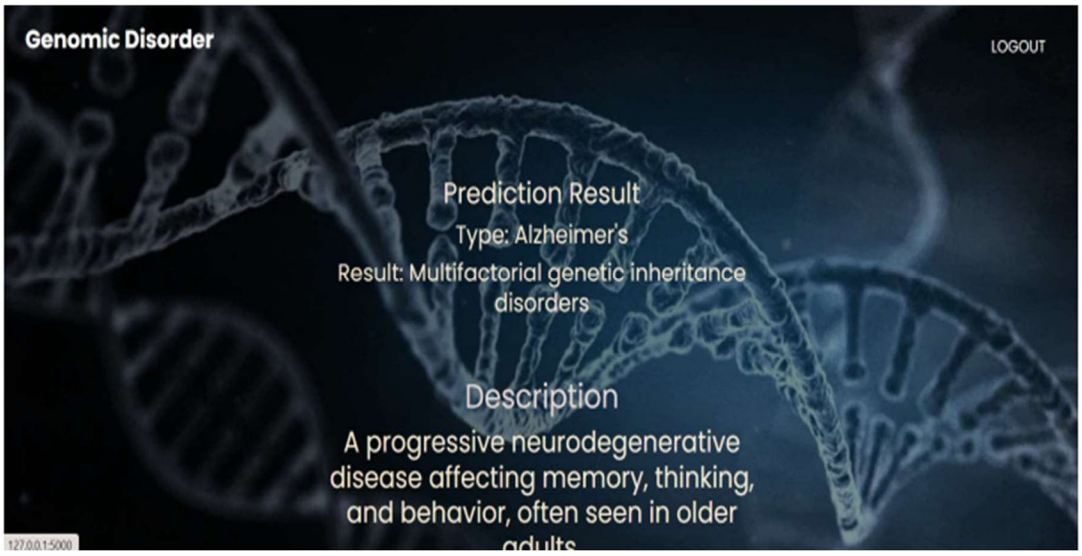


Figure 5.8 Predicted Outcome of Alzheimer's

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VI. CONCLUSION

A. *Summary*

A novel strategy is being developed to predict hereditary genetic disorders using machine learning models, offering more accurate and reliable predictions compared to traditional diagnostic methods. This system processes patient data through advanced preprocessing, feature engineering, and model training to deliver precise outcomes. It holds significant potential for early diagnosis, prevention, and management of genetic disorders, contributing to improved patient outcomes and better healthcare decision-making. The system continually evolves by integrating feedback and refining models iteratively, ensuring its ongoing relevance and effectiveness. By providing healthcare professionals with insightful data, the initiative showcases how data- driven approaches can revolutionize healthcare, particularly in genetic testing. Early prediction of genetic disorders—before birth or in childhood—opens up new opportunities for preventative care, allowing for timely interventions and informed treatment plans. Although the project demonstrates significant potential, it also presents challenges, such as the need for comprehensive, high-quality datasets and the complexities of refining machine learning models with real-world data. As healthcare technologies advance and access to medical data increases, these hurdles can be addressed. Overall, this system represents a significant step toward harnessing AI to enhance diagnostic accuracy, support preventative measures, and improve global healthcare outcomes.

B. *Future Enhancements*

1) Integration of Diverse and Comprehensive Datasets: By incorporating data from a wider range of medical conditions, genetic profiles, and environmental factors, the model's accuracy could be significantly improved.This could involve collaboration with hospitals, genetic research institutions, and healthcare providers to gather anonymized patient data.Integrating multi-modal data sources, such as genetic testing results, medical imaging, and family medical history, would provide a more holistic view of the patient's genetic predisposition, leading to more accurate and reliable predictions.

2) Development of Real-Time Prediction and Monitoring System: With the growing use of wearable health devices and real-time data collection, integrating these technologies into the platform could provide continuous genetic disorder risk assessments.This would enable early intervention and personalized treatment recommendations based on up-to-date patient data.

REFERENCES

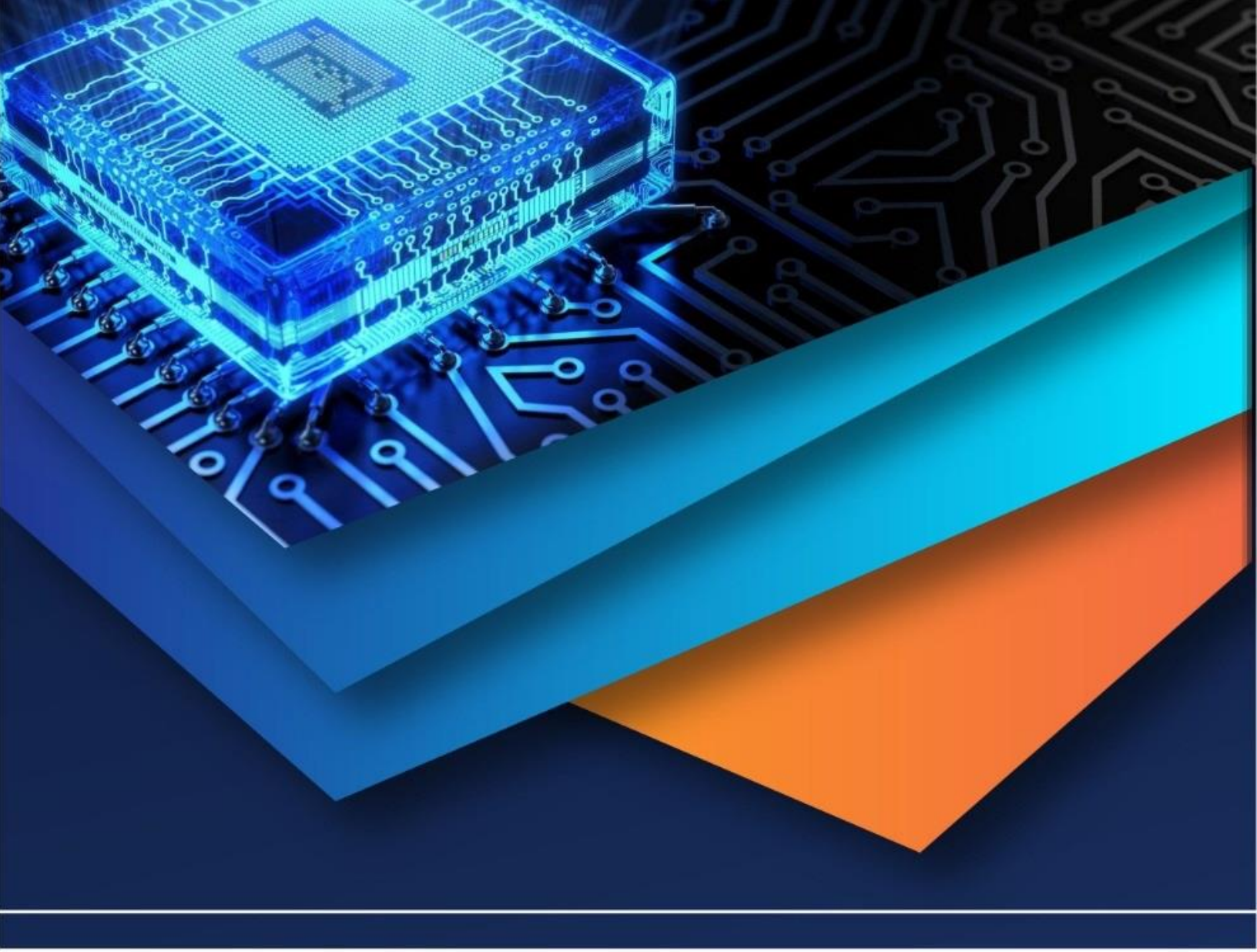
[1] L. Zhang, P. Li, and S. Wang, "A review on data preprocessing techniques in medical data mining," IEEE Transactions on Medical Imaging, vol. 40, no. 6, pp. 1450-1465, Jun. 2021.
[2] J. Smith, A. Brown, and R. Wilson, "A machine learning approach for predicting genetic disorders," Journal of Medical Genetics, vol. 45, no. 3, pp. 234-240, Mar. 2022.
[3] R. Clark and F. Harris, "Predictive models for hereditary diseases using ensemble learning techniques," Proceedings of the IEEE International Conference on Data Science and Machine Learning, pp. 450-455, Apr. 2022.
[4] N. Davis and T. White, "Feature selection methods in predictive healthcare models," Health Informatics Journal, vol. 27, no. 4, pp. 350-360, Nov. 2022.
[5] T. Nguyen, L. Green, and J. Parker, "Big data and machine learning in genetic disorder prediction," IEEE Access, vol. 10, pp. 55890-55902, Dec. 2022.
[6] M. Patel, R. Sharma, and D. Kumar, "Application of deep learning in genetic disorder prediction," International Journal of Health Informatics, vol. 18, no. 2, pp. 112-120, Feb. 2023.
[7] C. Johnson, K. Patel, and H. Lee, "Machine learning models for predicting genetic mutations in children," IEEE Transactions on Bioinformatics, vol. 39, no. 2, pp. 98-104, Feb. 2023.
[8] S. Thompson and J. Moore, "Genetic testing for rare diseases: A predictive model approach," Journal of Biomedical Informatics, vol. 35, no. 5, pp. 601-612, May 2023.
[9] A. Gupta, M. Desai, and K. Singh, "Improving genetic disorder diagnosis using machine learning," International Journal of Artificial Intelligence in Medicine, vol. 15, no. 1, pp. 65-75, Jan. 2024.
[10] J. Walker, P. Hall, and M. Rogers, "Implementing machine learning for genetic disorder prediction in clinical settings," Journal of Healthcare Technology, vol. 10, no. 3, pp. 180-192, Mar. 2024.
[11] Elavarasi T, Mariappan P, A deep learning approach to detect geneticbased disease in pregnancy period
[12] A Sangeetha, a., Ananthi, b. (2020). Genetic algorithm for feature selection to improve heart disease prediction by support vector machine. volume: 07 issue: 01, january 2020.
[13] Singh, S., Shukla, G., Agrawal, R., Dhule, C., Allabun, S., Alqahtani, M. S., Othman, M., Abbas, M., Soufiene, B. O. 2024. Enhancing Genomic Disorder Prediction Through Feynman Concordance And Interpolated Nearest Centroid Techniques. National Library of Medicine
[14] Rathod, S. V. K., Maruthiram, B. 2024. Advance genome disorder prediction model empowered with machine learning. IJCRT, 12(7), July 2024. ISSN: 2320-2882
[15] Janssens, A. C. J. W., van Duijn, C. M. (2009). Genome-based prediction of common diseases: Methodological considerations for future research. Genome Medicine
[16] Raza, A., Rustam, F., Siddiqui, H. U. R., de la Torre Diez, I., GarciaZapirain, B., Lee, E., Ashraf, I. (2022). Predicting genetic disorder and types of disorder using chain classifier approach. National Library of Medicine
[17] Atta-Ur-Rahman, M., Zubair, M., Nasir, M. U., Gollapalli, M., Saleem, M. A., Mehmood, S., Khan, M. A.,

[18] Mosavi, A. (2022). Advance genome disorder prediction model empowered with deep learning
[19] Sudha, V. P., M S, V. (2019). Deep learning based prediction of autism spectrum disorder using codon encoding of gene sequences. Journal Name, 9(1), October 2019. ISSN: 2249-8958
[20] Gayathri, T. T. (2017). Analysis of genomic sequences for prediction of cancerous cells using wavelet technique.April 201
[21] Sudha, V., Girijamma, D., Pragati, S. (2017). Classification of health disorder based on DNA technology. International Research Journal of Engineering and Technology (IRJET), 4(5), May 2017. p-ISSN: 2395- 0072
[22] Bele, A. D., Suryawanshi, V. K., Sharma, R. A., Deore, M. N. (2020). Cancer disease prediction using machine learning over big data. International Research Journal of Engineering and Technology , 7(3), March 2020. p-ISSN: 2395-0072
[23] Hemavathy, J., Jaya, B., Ananthi, T., Thomas, R. (2018). Disease identification using proteins values and regulatory modules. International Research Journal of Engineering and Technology (IRJET), 5(3), March 2018
[24] Singh, S. M., Hanchate, D. B. (2018). Improving disease prediction by machine learning. International Research Journal of Engineering and Technology (IRJET), 5(6), June 2018.
[25] Hemavathy, J., Jaya, B., Ananthi, T., Thomas, R. (year). Disease identification using proteins values and regulatory modules. International Research Journal of Engineering and Technology 5(3),2018

STUDENTS DETAILS

1	Name: SATHISH KUMAR GALI USN: 3VC21CS157 Phone No: 8792639848 Email-ID: skgali712@gmail.com Permanent Address: Opposite Udusalama Temple Indira Nagar , Brahmaappa street ,Badanahatti , Ballari (Dt) - 583116	
2	Name: NANDAKUMAR USN:3VC21CS107 Phone No: 9538733835 Email-ID: nandakumarnk349@gmail.com Permanent Address: 19th ward Gudneppanamatha kukanoor, koppal dist Karnataka -583232	
3	Name: MALLAIAH GARI SAI GANESH USN:3VC21CS096 Phone No: 6305691006 Email-ID: saiganeshmallaiahgari@gmail.com Permanent Address: Kadarampalli, Village and post Kundurpi, mandal Ananthapur, dist Andhra Pradesh, st, -515766	

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)