

DISEASE PREDICTION USING GENETIC MUTATION THROUGH BIG DATA ANALYSIS

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

VENNAPUSA SHANMUKENDAR REDDY (Reg.No-40111430)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Category – I University by UGC

Accredited “A++” by NAAC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
VENNAPUSA SHANMUKENDAR REDDY (Reg. No - 40111430) who carried
out the Project entitled "**DISEASE PREDICTION USING GENETIC MUTATION
THROUGH BIG DATA ANALYSIS**" under my supervision from NOVEMBER
2023 to April 2024.

G.I. 12/4/24
Internal Guide

Mrs. G. KAVITHA M.E, (PHD)

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for Viva voce Examination held on

12/9/24

Refer
Internal Examiner

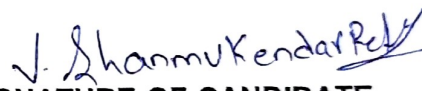
External Examiner

DECLARATION

I, **VENNAPUSA SHANMUKENDAR REDDY** (Reg. No - 40111430) hereby declare that the Project Report entitled "**DISEASE PREDICTION USING GENETIC MUTATION THROUGH BIG DATA ANALYSIS**" done by me under the guidance of **Mrs. G. KAVITHA M.E, (PHD)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 12-04-2024

PLACE: CHENNAI


SIGNATURE OF CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. G. KAVITHA M.E, (PHD)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In the Existing system, Gene dependency networks often undergo changes with respect to different disease states. Understanding how these networks rewire between two different disease states is an important task in genomic research. In the Proposed system, a new differential network inference model which identifies gene network rewiring by combining gene expression and gene mutation data. Similarities and differences between different data types are learned via a group bridge penalty function. In the Modification process, project is to recommend Diet pattern or any other natural drugs which can be recommended to those people who is expected to get into the disease in a course of time by verifying the mutated Genes. In the era of precision medicine, leveraging big data analytics to predict disease susceptibility based on genetic mutations has become increasingly important. This abstract presents a concise overview of the process involved in disease prediction through genetic mutation analysis using big data. Beginning with the collection of vast genetic datasets from diverse sources, the data undergoes rigorous preprocessing to ensure accuracy and consistency. Feature extraction techniques identify pertinent genetic mutations associated with various diseases. Machine learning models, including supervised, unsupervised, and deep learning approaches, are employed to analyze the complex relationships between genetic mutations and disease outcomes. Model performance is evaluated using standard metrics, and ethical considerations are addressed to ensure patient privacy and consent. Ultimately, the application of predictive models in clinical settings enables personalized medicine approaches tailored to individual genetic profiles, thereby enhancing disease prevention and treatment strategies. This abstract emphasizes the interdisciplinary nature of this research, requiring collaboration between bioinformatics, genetics, medicine, and computer science to realize the full potential of disease prediction through genetic mutation analysis using big data.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	V
	LIST OF FIGURES	Vii
	LIST OF ABBREVIATIONS	Viii
01	INTRODUCTION	1
	1.1 Domain introduction	2
02	LITERATURE SURVEY	13
	2.1 Inferences From Literature Survey And Open Problem In Existing System	14
03	REQUIREMENTS ANALYSIS	19
	3.1 Feasibility Studies/Risk Analysis Of The Project	19
	3.2 Software Requirements Specification Document	20
	3.3 System Use case	21
04	DESCRIPTION OF PROPOSED SYSTEM	27
	4.1 Selected Methodology Or Process Model	28
	4.2 Architecture/Overall Design Of Proposed System	32
	4.3 Data Flow Diagram	33
	4.4 Description Of Software For Implementation And Testing Plan Of The Proposed Model/System	35
05	IMPLEMENTATION DETAILS	46
	5.1 Development And Deployment Setup	46
	5.2 Algorithms	48
	5.3 Testing	49
06	RESULTS AND DISCUSSION	50
	6.1 Results	50
	6.2 Discussion	51
07	CONCLUSION	52
	7.1 Conclusion	52
	7.2 FUTURE WORK	53
	7.3 IMPLEMENTATION ISSUES	54
	REFERENCES	56
	APPENDIX	57
	A Source Code	57
	B Screenshots	63
	C RESEARCH PAPER	76

LIST OF FIGURES

Figure No.	FIGURE NAME	Page No.
1.1	Four V's Of Big Data	3
1.2	Transformation Process	8
3.1	Use Case Diagram	22
3.2	Class Diagram	23
3.3	Sequence Diagram	24
3.4	Collaboration Diagram	25
3.5	Activity Diagram	26
4.1	User Interface Design Diagram	29
4.2	Data Set Maintenance Diagram	29
4.3	Outer Layer Removal Diagram	30
4.4	Disease Prediction Diagram	30
4.5	Drug Suggestion Diagram	30
4.6	System Architecture	31
4.7	System Design Level 0	32
4.8	System Design Level 1	33
4.9	Software Development Process	38
4.10	API and JVM	39
5.1	Taxonomy Of Testing	48

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
DP	Disease Prediction
GM	Genetic Mutation
BDA	Big Data Analysis
GMDP	Genetic Mutation Disease Prediction
BDGMA	Big Data Genetic Mutation Analysis
DGM	Disease Genetic Mutation
DPBDA	Disease Genetic Mutation Big Data Analysis
GMBDA	Genetic Mutation Big Data Analysis

CHAPTER 1

INTRODUCTION

Characterizing gene dependency networks can help to understand the causes and mechanisms of diseases. It is well established that gene dependency networks often undergo changes with respect to different disease states. The development and progression of cancers often involve aberrations in gene networks. Therefore, inferring the differential networks between two disease states from high throughput data, which can reveal the structure changes of gene networks, is of great interest in the field of bioinformatics. Differential networks are often inferred from gene expression data based on Gaussian Markov networks. By assuming that the gene expression measurements follow a multivariate Gaussian distribution, the conditional dependencies among genes can be directly determined by the nonzero elements of the precision matrix (the inverse covariance matrix). Two genes are conditionally independent given the other genes if and only if the corresponding element of the precision matrix is zero. Therefore, the state-specific gene networks can be modeled by the corresponding precision matrices, and the differential network between two states can be modeled as the difference between the two state-specific precision matrices. The most straightforward one is to separately estimate the state-specific precision matrices first and then compute their difference (called indirect estimation methods). This type of methods often applies sparse penalties to the individual precision matrices to deal with the high dimensional cases where the number of genes is larger than the number of observations. This would be suboptimal when the individual precision matrices are not sparse but only their difference is sparse. The other type of approaches is to directly estimate the precision matrix difference without estimating the individual state-specific precision matrices (called direct estimation methods). Compared with the indirect estimation methods, the direct methods have an advantage of using much smaller sample size to achieve competitive performance.

1.1 DOMAIN INTRODUCTION

Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications. The challenges include analysis, capture, duration, search, sharing, storage, transfer, visualization, and privacy violations. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, prevent diseases, combat crime and so on. So, we can implement big data in our project because every employ has instructed information so we can make analysis on this data.

1.1.1 Characteristics Of Big Data

Big data is a term utilized to refer to the increase in the volume of data that are difficult to store, process, and analyze through traditional database technologies. The nature of big data is indistinct and involves considerable processes to identify and translate the data into new insights. The term "big data" is relatively new in IT and business. However, several researchers and practitioners have utilized the term in previous literature. For instance, referred to big data as a large volume of scientific data for visualization. Several definitions of big data currently exist." Meanwhile and defined big data as characterized by three Vs: volume, variety, and velocity. The terms volume, variety, and velocity were originally introduced by Gartner to describe the elements of big data challenges. IDC also defined big data technologies as "a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling the high velocity capture, discovery, and/or analysis." specified that big data is not only characterized by the three Vs mentioned above but may also extend to four Vs, namely, volume, variety, velocity, and value This 4V definition is widely recognized because it highlights the meaning and necessity of big data.



Fig 1.1 Four V's of Big Data

Volume - Refers to the amount of all types of data generated from different sources and continue to expand. The benefit of gathering large amounts of data includes the creation of hidden information and patterns through data analysis. Collecting longitudinal data requires considerable effort and underlying investments. Nevertheless, such mobile data challenge produced an interesting result similar to that in the examination of the predictability of human behavior patterns or means to share data based on human mobility and visualization techniques for complex data.

Variety - Refers to the different types of data collected via sensors, smart phones, or social networks. Such data types include video, image, text, audio, and data logs, in either structured or unstructured format. Most of the data generated from mobile applications are in unstructured format. For example, text messages, online games, blogs, and social media generate different types of unstructured data through mobile devices and sensors.

Velocity - Refers to the speed of data transfer. The contents of data constantly change because of the absorption of complementary data collections, introduction of previously archived data or legacy collections, and streamed data arriving from multiple sources.

Value - is the most important aspect of big data; it refers to the process of discovering huge hidden values from large datasets with various types and rapid generation.

1.1.2. Classification of Big Data

Big data are classified into different categories to better understand their characteristics shows the numerous categories of big data. The classification is important because of large-scale data in the cloud. The classification is based on five aspects: (i) data sources, (ii) content format, (iii) data stores, (iv) data staging, and (v) data processing. Data sources include internet data, sensing and all stores of transnational information, ranges from unstructured to highly structured are stored in various formats. Most popular is the relational database that come in a large number of varieties. As the result of the wide variety of data sources, the captured data differ in size with respect to redundancy, consistency.

i. Big Data Storage System

The rapid growth of data has restricted the capability of existing storage technologies to store and manage data. Over the past few years, traditional storage systems have been utilized to store data through structured RDBMS. However, almost storage systems have limitations and are inapplicable to the storage and management of big data. A storage architecture that can be accessed in a highly efficient manner while achieving availability and reliability is required to store and manage large datasets. Several storage technologies have been developed to meet the demands of massive data. Existing technologies can be classified as direct attached storage (DAS), network attached storage (NAS), and storage area network (SAN). In DAS, various hard disk drives (HDDs) are directly connected to the servers. Each HDD receives a certain amount of input/output (I/O) resource, which is managed by individual applications. Therefore, DAS is suitable only for servers that are interconnected on a small scale. Given the aforesaid low scalability, storage capacity is increased but expandability and upgradeability are limited significantly. NAS is a storage device that supports a network.

Given that the NAS server can indirectly access a storage device through networks, the I/O burden on a NAS server is significantly lighter than that on a DAS server. NAS can orient networks, particularly scalable and bandwidth-intensive networks. Such networks include high-speed networks of optical-fiber connections. The SAN system of data storage is independent with respect to storage on the local area network (LAN). Multipath data switching is conducted among internal nodes to maximize data management and sharing. The organizational systems of data storages (DAS, NAS, and SAN) can be divided into three parts: (i) disc array, where the foundation of a storage system provides the fundamental guarantee, (ii) connection and network subsystems, which connect one or more disc arrays and servers, and (iii) storage management software, which oversees data sharing, storage management, and disaster recovery tasks for multiple servers.

ii. Hadoop Background

Hadoop is an open-source Apache Software Foundation project written in Java that enables the distributed processing of large datasets across clusters of commodity. Hadoop has two primary components, namely, HDFS and Map Reduce programming framework. The most significant feature of Hadoop is that HDFS and Map Reduce are closely related to each other; each are co-deployed such that a single cluster is produced. Therefore, the storage system is not physically separated from the processing system.

HDFS is a distributed file system designed to run on top of the local file systems of the cluster nodes and store extremely large files suitable for streaming data access. HDFS is highly fault tolerant and can scale up from a single server to thousands of machines, each offering local computation and storage. HDFS consists of two types of nodes, namely, a name node called “master” and several data nodes called “slaves.” HDFS can also include secondary name nodes. The name node manages the hierarchy of file systems and director namespace (i.e., metadata).

Map Reduce is a simplified programming model for processing large numbers of datasets pioneered by Google for data-intensive applications. The Map Reduce model was developed based on GFS is adopted through open-source Hadoop implementation, which was popularized by Yahoo. Apart from the Map Reduce framework, several other current open-source Apache projects are related to the Hadoop ecosystem, including Hive, Mahout. An overview of current Map Reduce projects and related software is shown in Map Reduce allows an inexperienced programmer to develop parallel programs and create a program capable of using computers in a cloud. The mapper regards the key/value pair as input and generates intermediate key/value pairs. The reducer merges all the pairs associated with the same (intermediate) key and then generates an output. Summarizes the process of the map/reduce function.

iii. Map Reduce in Clouds

Map Reduce accelerates the processing of large amounts of data in a cloud; thus, Map Reduce, is the preferred computation model of cloud providers. Map Reduce is a popular cloud computing framework that robotically performs scalable distributed applications and provides an interface that allows for parallelization and distributed computing in a cluster of servers. The approaches to apply scientific computing problems to the Map Reduce framework where scientists can efficiently utilize existing resources in the cloud to solve computationally large-scale scientific data.

iv. Research Challenges

Although cloud computing has been broadly accepted by many organizations, research on big data in the cloud remains in its early stages. Several existing issues have not been fully addressed. Moreover, new challenges continue to emerge from applications by organization. In the subsequent sections, some of the key research challenges, such as scalability, availability, data integrity, data transformation, data quality, data heterogeneity, privacy and legal issues, and regulatory governance, are discussed.

v. Scalability

Scalability is the ability of the storage to handle increasing amounts of data in an appropriate manner. Scalable distributed data storage systems have been a critical part of cloud computing infrastructures. The lack of cloud computing features to support RDBMSs associated with enterprise solutions has made RDBMSs less attractive for the deployment of large-scale applications in the cloud. This drawback has resulted in the popularity of NoSQL

A NoSQL database provides the mechanism to store and retrieve large volumes of distributed data. The features of NoSQL databases include schema-free, easy replication support, simple API, and consistent and flexible modes. Different types of NoSQL databases, such as key-value column-oriented, and document-oriented, provide support for big data. Shows a comparison of various NoSQL database technologies that provide support for large datasets.

Vi. Availability

Availability refers to the resources of the system accessible on demand by an authorized individual. In a cloud environment, one of the main issues concerning cloud service providers is the availability of the data stored in the cloud. For example, one of the pressing demands on cloud service providers is to effectively serve the needs of the mobile user who requires single or multiple data within a short amount of time. Therefore, services must remain operational even in the case of a security breach. In addition, with the increasing number of cloud users, cloud service providers must address the issue of making the requested data available to users to

deliver high-quality services. Lee et al. introduced a multi-cloud model called “rain clouds” to support big data exploitation. “Rain clouds” involves cooperation among single clouds to provide accessible resources in an emergency. Schroeck et al. predicted that the demand for more real time access to data may continue to increase as business models evolve and organizations invest in technologies required for streaming data and smart phones.

vi. Data Integrity

A key aspect of big data security is integrity. Integrity means that data can be modified only by authorized parties or the data owner to prevent misuse. The proliferation of cloud-based applications provides users the opportunity to store and manage their data in cloud data centers. Such applications must ensure data integrity. However, one of the main challenges that must be addressed is to ensure the correctness of user data in the cloud.

vii. Transformation

Transforming data into a form suitable for analysis is an obstacle in the adoption of big data. Owing to the variety of data formats, big data can be transformed into an analysis workflow in two ways.

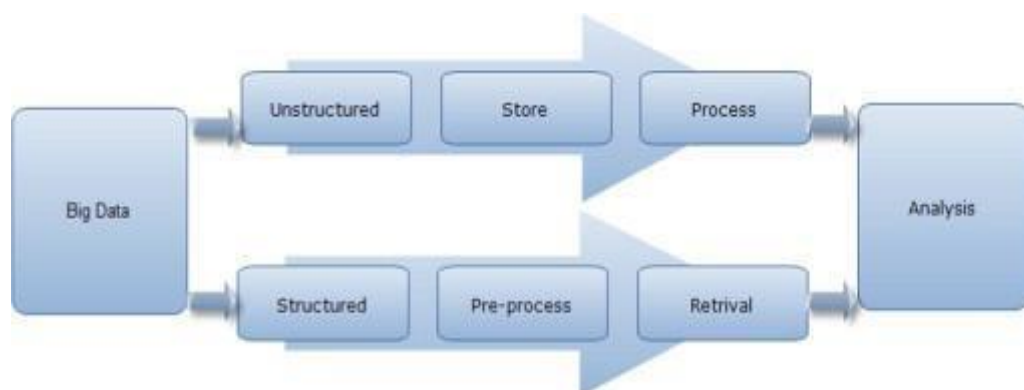


Fig 1.2 Transformation process

viii. Data Quality

In the past, data processing was typically performed on clean datasets from well-known and limited sources. Therefore, the results were accurate. However, with the emergence of big data, data originate from many different sources; not all of these sources are well-known or verifiable. Poor data quality has become a serious problem for many cloud service providers because data are often collected from different sources. For example, huge amounts of data are generated from smart phones, where inconsistent data formats can be produced as a result of heterogeneous sources. The data quality problem is usually defined as “any difficulty encountered along one or more quality dimensions that render data completely or largely unfit for use”. Therefore, obtaining high-quality data from vast collections of data sources is a challenge. High-quality data in the cloud is characterized by data consistency. If data from new sources are consistent with data from other sources, then the new data are of high quality.

ix. Heterogeneity

Variety, one of the major aspects of big data characterization, is the result of the growth of virtually unlimited different sources of data. This growth leads to the heterogeneous nature of big data. Data from multiple sources are generally of different types and representation forms and significantly interconnected; they have incompatible formats and are inconsistently represented.

x. Privacy

Privacy concerns continue to hamper users who outsource their private data into the cloud storage. This concern has become serious with the development of big data mining and analytics, which require personal information to produce relevant results, such as personalized and location-based services. Information on individuals is exposed to scrutiny, a condition that gives rise to concerns on profiling, stealing, and loss of control.

xi. Legal/Regulatory Issues

Specific laws and regulations must be established to preserve the personal and sensitive information of users. Different countries have different laws and regulations to achieve data privacy and protection. In several countries, monitoring of company staff communications is not allowed. However, electronic monitoring is permitted under special circumstances. Therefore, the question is whether such laws and regulations offer adequate protection for individuals' data while enjoying the many benefits of big data in the society at large.

xii. Governance

Data governance embodies the exercise of control and authority over data-related rules of law, transparency, and accountabilities of individuals and information systems to achieve business objectives. The key issues of big data in cloud governance pertain to applications that consume massive amounts of data streamed from external sources. Therefore, a clear and acceptable data policy with regard to the type of data that need to be stored, how quickly an individual needs to access the data, and how to access the data must be defined.

Big data governance involves leveraging information by aligning the objectives of multiple functions, such as telecommunication carriers having access to vast troves of customer information in the form of call detail records and marketing seeking to monetize this information by selling it to third parties.

xiii. Open Research Issues

Numerous studies have addressed a number of significant problems and issues pertaining to the storage and processing of big data in clouds. The amount of data continues to increase at an exponential rate, but the improvement in the processing mechanisms is relatively slow. Only a few tools are available to address the issues of big data processing in cloud environments. State-of-the-art techniques and technologies in many important

big data applications (i.e., MapReduce, Dryad, Pregel, PigLatin, MangoDB, Hbase, SimpleDB, and Cassandra) cannot solve the actual problems of storing and querying big data. For example, Hadoop and Map Reduce lack query processing strategies and have low-level infrastructures with respect to data processing and management. Despite the plethora of work performed to address the problem of storing and processing big data in cloud computing environments, certain important aspects of storing and processing big data in cloud computing are yet to be solved. Some of these issues are discussed in the subsequent subsections.

xiv. Data Staging

The most important open research issue regarding data staging is related to the heterogeneous nature of data. Data gathered from different sources do not have a structured format. For instance, mobile cloud-based applications, blogs, and social networking are inadequately structured similar to pieces of text messages, videos, and images. Transforming and cleaning such unstructured data before loading those into the warehouse for analysis are challenging tasks. Efforts have been exerted to simplify the transformation process by adopting technologies such as Hadoop and Map Reduce to support the distributed processing of unstructured data formats.

xv. Distributed Storage Systems

Numerous solutions have been proposed to store and retrieve massive amounts of data. Some of these solutions have been applied in a cloud computing environment. However, several issues hinder the successful implementation of such solutions, including the capability of current cloud technologies to provide necessary capacity and high performance to address massive amounts of data, optimization of existing file systems for the volumes demanded by data mining applications, and how data can be stored in such a manner that they can be easily retrieved and migrated between servers.

xvi. Data Analysis

The selection of an appropriate model for large-scale data analysis is critical. Talia pointed out that obtaining useful information from large amounts of data requires scalable analysis algorithms to produce timely results. However, current algorithms are inefficient in terms of big data analysis. Therefore, efficient data analysis tools and technologies are required to process such data. Each algorithm performance ceases to increase linearly with increasing computational resources. As researchers continue to probe the issues of big data in cloud computing, new problems in big data processing arise from the transitional data analysis techniques. The speed of stream data arriving from different data sources must be processed and compared with historical information within a certain period of time. Such data sources may contain different formats, which makes the integration of multiple sources for analysis a complex task.

xvii. Data Security

Although cloud computing has transformed modern ICT technology, several unresolved security threats exist in cloud computing. These security threats are magnified by the volume, velocity, and variety of big data. Moreover, several threats and issues, such as privacy, confidentiality, integrity, and availability of data, exist in big data using cloud computing platform.

CHAPTER 2

Literature Survey

Disease prediction methods based on ML & DM have been employed by many researchers in the medical field. Some of these are discussed in further detail below. Data from 1,546 workers' yearly health examinations were evaluated by Suzuki et al.. After 12 months of systematic health checks to assess lifestyle modifications with such a shift in blood transaminase levels, they found that in order to cure alcohol content fatty liver disease, it is recommended that patients lose 6% of their body mass by calorie restriction and regular exercise. After stabilizing their ALT levels, 136 patients were monitored for a full year to see whether there was a correlation between the changes they made to their lifestyle and their ALT readings. Their findings proved that a healthy lifestyle shift may reverse nonalcoholic fatty liver disease. They have also examined the usefulness of ML and graph theory measures for identifying Parkinson's illness. An intelligent system was suggested by S. A. Patte kari and A. Praveen that employs a DM approach to capture user responses for specified questions relating to sugar levels, gender, weight, etc. and match them to recorded information results, i.e., trained dataset. Data from surveys claiming 75% accuracy were used in a discussion between A. Annand and D. Shakti concerning the association between a person's eating habits, sleeping patterns, level of physical activity, and body mass index and the likelihood of developing diabetes. one technique, which is simple to train. Data mining (DM) methods were used by Hossain et al. to make predictions about the prevalence of obesity risk factors and to conduct statistical data exploration in order to identify the population in Bangladesh most negatively impacted by obesity.

2.1 INFERENCES FROM LITERATURE SURVEY AND OPEN PROBLEMS IN EXISTING SYSTEM

I. Paper 1: Developing an Index for Detection and Identification of Disease Stages (2019)

Authors: David Ashourlo, Ali Akbar Matkan, Alfredo Huete, Hossein Aghighi, and Mohammad Reza Mobasheri

Abstract: Spectral data have been widely used to estimate the disease severity (DS) levels of different plants. However, such data have not been evaluated to estimate the disease stages of the plant. This study aimed at developing a spectral disease index (SDI) that is able to identify the stages of wheat leaf rust disease at various DS levels. To meet the aim of the study, the reflectance spectra (350–2500 nm) of infected leaves with different symptom fractions and DS levels were measured with a spectro radio meter. Then, pure spectra of the different disease symptoms at the leaf scale were analyzed, and a new function was developed to find the wavelengths most sensitive to disease symptom fraction. The reflectance spectra with highest sensitivity were found at 675 and 775 nm. Finally, the normalized difference of DS and the ratio p_{675}/p_{775} was used as a new SDI to discriminate three different levels of the disease stage at the canopy level. The suggested SDI showed a promising performance to improve the detection disease stages in precision plant protection

Inference: The experiment was performed on Plants for it Disease Prediction. It is implemented to identify three different stages of WLR disease. In stage 1, the leaf begins to show symptoms with yellow and orange colors predominant. In stage 2, all of the symptoms are fully developed and can be seen with similar proportion of colors. Stage 3 includes dry leaves in which the predominant symptoms show as brown color.

Disadvantages: It is not for Human Disease Diagnosis, It is not implemented on Genome based diagnosis. It is not a Prediction / Early detection of Disease Detection process. Big data is not Used.

Paper 2: Identifying The Candidate Genes for Alzheimer's Disease Based On The Rejection Region Of T Test (2020)

Authors: GUI-QIONG ZHU¹, PEI-HUI YANG²

Abstract: Modern medical research has proved that almost all diseases are related to genes except the injuries. Based on the Alzheimer's disease (AD) gene expression data, this paper uses T test method to identify the significant differences in gene expression between normal people and three states of Alzheimer's disease patients. In this paper, 90 differentially expressed genes were identified, and 5 of which have been confirmed to be associated with Alzheimer's disease by other references, 10 of which are associated with nerve cell tissue and signaling. This result indicates that the identified genes (at least some of them) are likely to be related to Alzheimer's Disease

Inference: Gene Sequence Analysis towards Disease Diagnosis (Alzheimer's disease) using K mean & PCA Algorithms.

Disadvantages: Just a diagnosis Tool and not working towards Prevention mechanism to others. No Medical Data is used. Big data is not Used.

II. Paper 3: A New Method For Disease-Related gene Prioritization (2021)

Author: Yaogong Zhang

Abstract: Prioritizing genes according to their association with a disease allows researchers to explore genes in more informed ways. Although some useful algorithms have been developed, they are based on single gene importance, gene interaction networks, or gene modules with little consideration of relative gene importance in the context of modules. In this paper, we propose to prioritize genes considering both individual genes and their affiliated modules, and utilize Gene Ontology (GO) based fuzzy measure value as well as known disease genes as heuristics. The performance of our method is comprehensively validated by using both simulated and real datasets. Results show that our method outperforms other methods in terms of disease-related gene prioritization. This work will aid researchers in the understanding of the genetic architecture of complex diseases, and improve the accuracy of diagnosis and the effectiveness of therapy.

Inference: This Paper, prioritize genes considering both individual genes and their affiliated modules, and utilize Gene Ontology (GO) based fuzzy measure value as well as known disease genes as heuristics.

Disadvantages: Polymorphism is not considered in analysis, Big data is not used. Simulated data set is used for their advantage, so Result may not be considered Genuine also.

III. Paper4 Crledd: Regularized Causalities Learning for Early Detection of Diseases Using Electronic Health Record (Ehr) Data (2021)

Authors: Jiang Bian, Sijia Yang, Haoyi Xiong, Licheng Wang, Yanjie Fu, Zeyi Sun, Zhishan Guo, and Jun Wang.

Abstract: The availability of Electronic Health Records (EHR) in health care settings has provided tremendous opportunities for early disease detection. While many supervised learning models have been adopted for EHR-based disease early detection, the ill-posed inverse problem in the parameter learning has imposed a significant challenge on improving the accuracy of these algorithms. In this paper, we propose CRLEDD – Causality-Regularized Learning for Early Detection of Disease, an algorithm to improve the performance of Linear Discriminant Analysis (LDA) on top of diagnosis-frequency vector data representation. While most existing regularization methods exploit sparsity regularization to improve detection performance, CRLEDD provides a unique perspective by ensuring positive semi- definiteness of the sparsified precision matrix used in LDA which is different from the regular regularization method (e.g. L2regularization).To achieve this goal, CRLEDD employs Graphical Lasso to estimate the precision matrix in the ill-posed settings for enhanced accuracy of LDA classifiers. We perform extensive evaluation of CRLEDD using a large-scale real-world EHR dataset to predict mental health disorders (e.g., depression and anxiety) of college students from 10 universities in the U.S. We compare CRLEDD with other regularized LDA and downstream classifiers.

Inference: Given EHR data represented as vectors and graphs, researchers have proposed to predict the target disease through supervised learning, using downstream classifiers.

Disadvantages: It is not applied on genome Sequence. It is applied only on PHR Medical data.

IV. Paper 5: Diagnosis Of Kashin-Beck Disease And Other Common Joint Diseases Via A Gene Model (2022)

Author: Yanan Zhang

Abstract: Kashin-Beck disease and common joint disease display similar early clinical features, result in the difficulty of identification of diagnosis and treatment by observing the clinical manifestations and physical signs. In this paper, we use two sets of gene expression data from cartilaginous tissues and a health tissues, collected from 16 samples. We propose a gene model to achieve the diagnosis of Kashin-Beck disease and other common joint diseases using the expression. In the model, we try to filter various noises caused by improper operation, aging of equipment and so on by wavelet transform. Bhattacharyya distance and Pearson correlation coefficient are used to remove irrelevant genes and redundant genes respectively and acquire the rest as feature genes. In order to deal with the comprehensive data, we choose support vector machine (SVM) to achieve identification. The experiment results demonstrate that the gene model proposed in this paper offers a set of feature genes for further biological interpretation and play an important role in the diagnosis.

Inference: Gene model to achieve the diagnosis of Kashin-Beck disease and other common joint diseases using the expression

Disadvantages: Big Data is not used in this Project. Not spoken about Large Data set. This project is not aimed at Early Discovery of Disease and not working towards Prevention process.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

3.1.1 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- OPERATIONAL FEASIBILITY

I. Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

II. Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources system.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

I. Hardware specification:

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

Operating system	:	Windows 7/
8.1 HDD	:	500GB
RAM	:	2-4GB
Processor	:	Core i3, i5

II. Software specification:

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

Front End	:	JDK 7/8
Back End	:	MySQL
IDE	:	NetBeans 8.1 / 8.2
Tool	:	Hadoop

3.3 SYSTEM USECASE

3.3.1. Uml Diagrams:

The Unified Modeling Language (UML) was created to forge a common, semantically and syntactically rich visual modeling language for the architecture, design, and implementation of complex software systems both structurally and behaviorally. UML has applications beyond software development, such as process flow in manufacturing.

It is analogous to the blueprints used in other fields, and consists of different types of diagrams. In the aggregate, UML diagrams describe the boundary, structure, and the behavior of the system and the objects within it.

UML is not a programming language but there are tools that can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design.

3.3.2. Use Case Diagrams:

A use case is a list of steps that define interaction between an actor (a human who interacts with the system or an external system) and the system itself. Use case diagrams depict the specifications of a use case and model the functional units of a system. These diagrams help development teams understand the requirements of their system, including the role of human interaction therein and the differences between various use cases. A use case diagram might display all use cases of the system, or just one group of use cases with similar functionality.

1. To begin a use case diagram, add an oval shape to the center of the drawing.
2. Type the name of the use case inside the oval.
3. Represent actors with a stick figure near the diagram, then use lines to model relationships between actors and use cases.

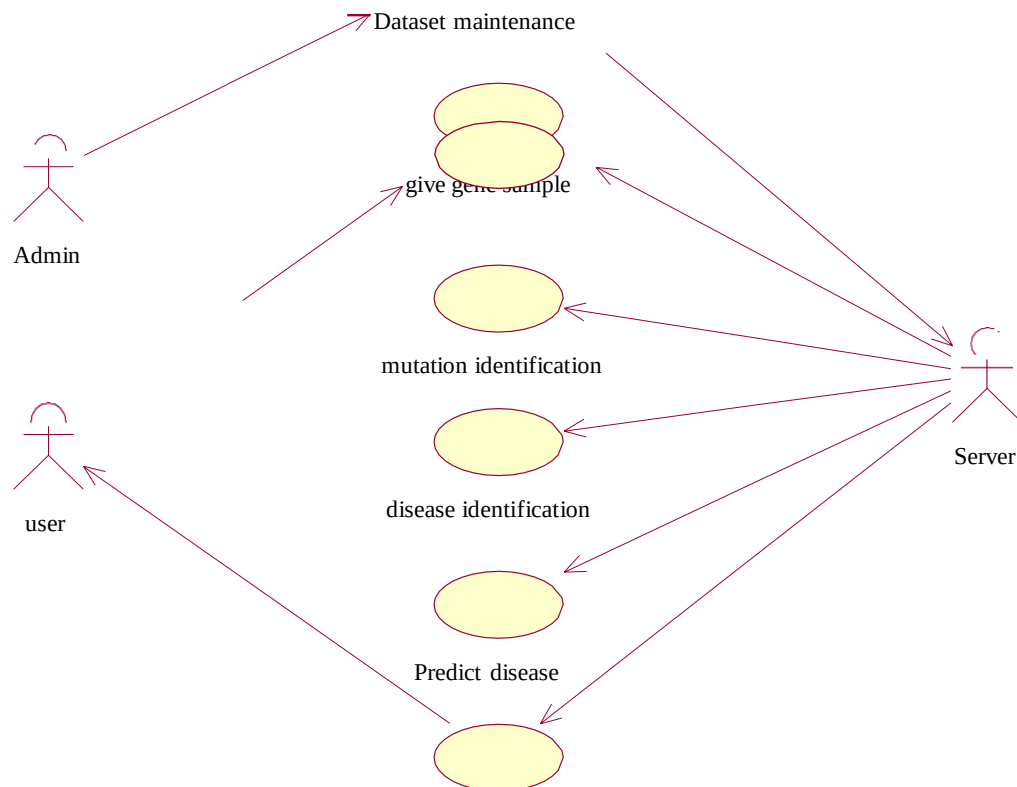


Fig. No: 3.1 Use Case Diagram

I. Class Diagram:

Class diagrams represent the static structures of a system, including its classes, attributes, operations, and objects. A class diagram can display computational data or organizational data in the form of implementation classes and logical classes, respectively. There may be overlap between these two groups.

1. Classes are represented with a rectangular shape that is split into thirds. The top section displays the class name, while the middle section contains the class' attributes. The bottom section features the class operations (also known as methods).

2. Add class shapes to your class diagram to model the relationship between those objects. You may need to add subclasses, as well.
3. Use lines to represent association, inheritance, multiplicity, and other relationships between classes and subclasses. Your preferred notation style will inform the notation of these lines.

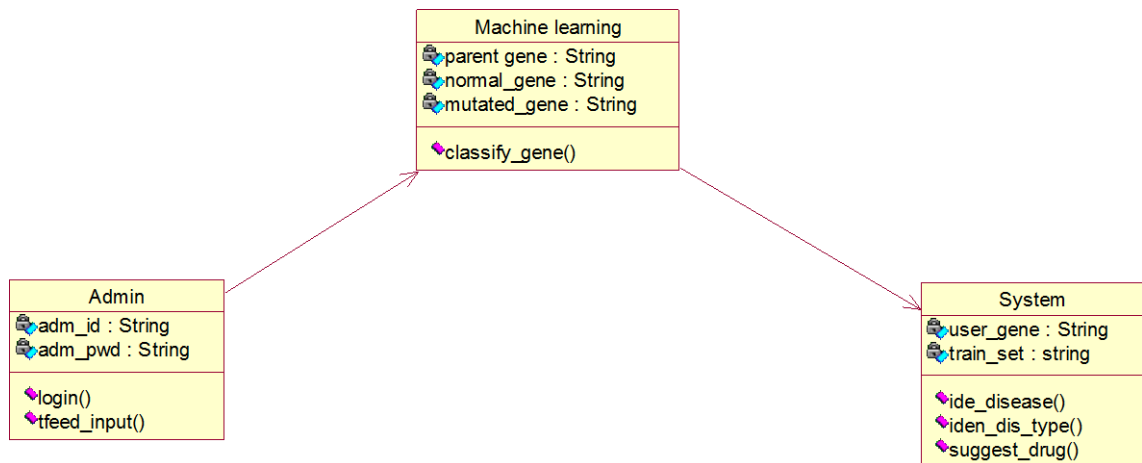


Fig. No: 3.2 Class Diagram

II. Sequence Diagram:

Sequence diagrams, also known as event diagrams or event scenarios, illustrate how processes interact with each other by showing calls between different objects in a sequence. These diagrams have two dimensions: vertical and horizontal. The vertical lines show the sequence of messages and calls in chronological order, and the horizontal elements show object instances where the messages are relayed.

1. To create a sequence diagram, write the class instance name and class name in a rectangular box.
2. Draw lines between class instances to represent the sender and receiver of messages.
3. Use solid arrowheads to symbolize synchronous messages, open arrowheads for asynchronous messages, and dashed lines for reply messages.

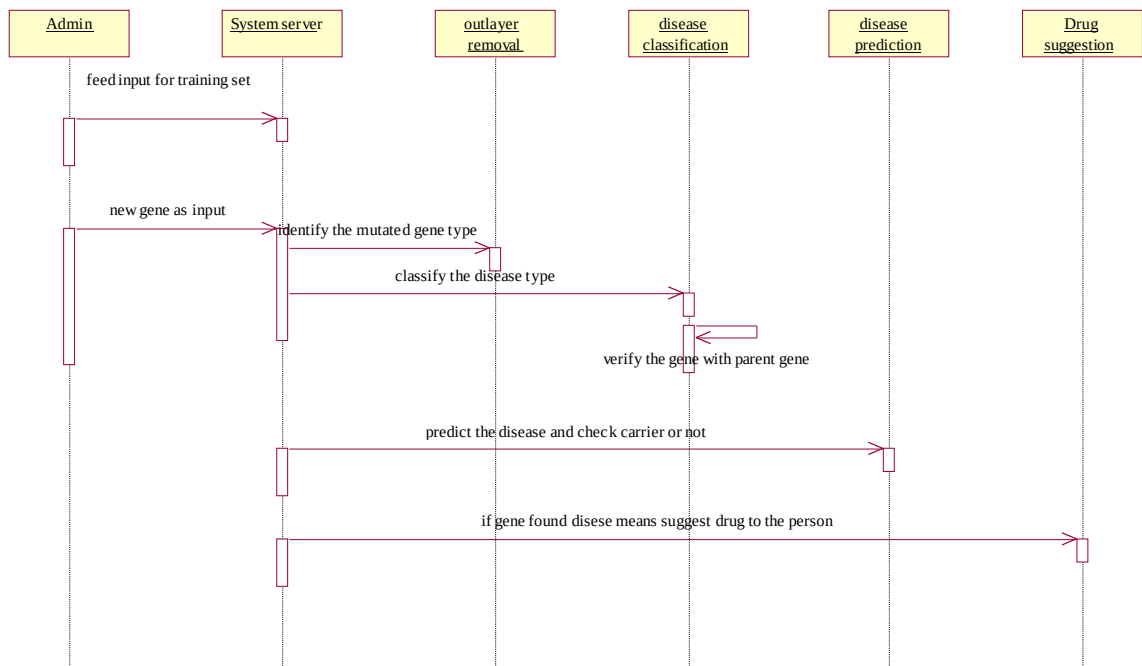


Fig. No: 3.3: Sequence Diagram

III. Collaboration Diagram:

Communication diagrams offer benefits similar to sequence diagrams, but they will offer a better understanding of how components communicate and interact with each other rather than solely emphasizing the sequence of events. They can be a useful reference for businesses, organizations, and engineers who need to visualize and understand the physical communications within a program. Try drawing a sequence diagram to:

- Model the logic of a sophisticated procedure, function, or operation.
- Identify how commands are sent and received between objects or components of a process.
- Visualize the consequences of specific interactions between various components in a process.
- Plan and understand the detailed functionality of an existing or future scenario.

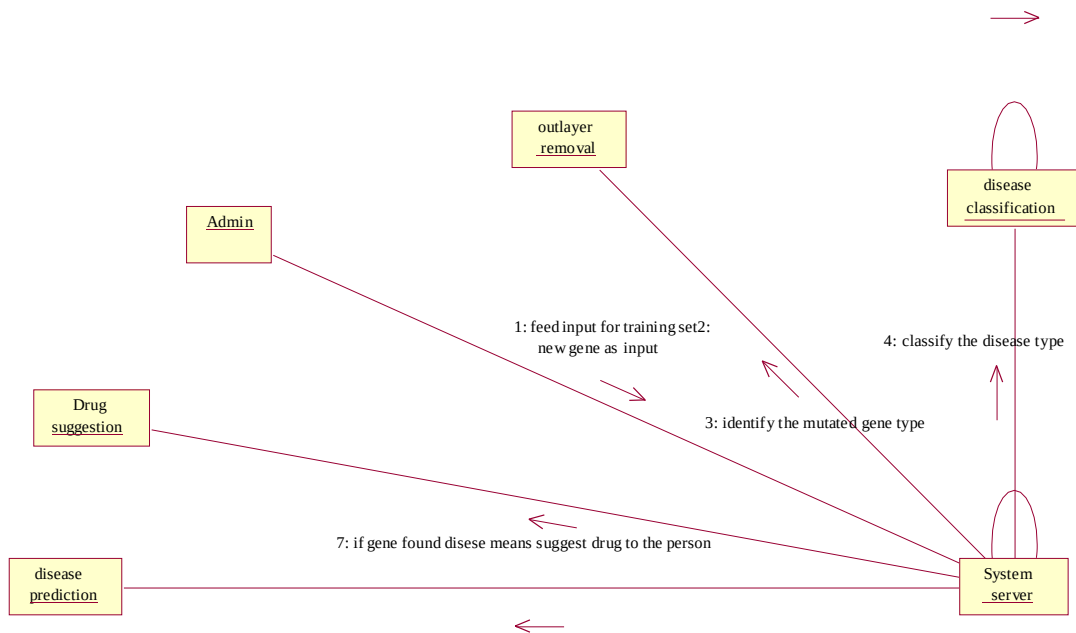


Fig. No: 3.4 Collaboration Diagram

IV. Activity Diagram:

Activity diagrams show the procedural flow of control between class objects, along with organizational processes like business workflows. These diagram are made of specialized shapes, then connected with arrows. The notation set for activity diagrams is similar to those for state diagrams.

1. Begin your activity diagram with a solid circle.
2. Connect the circle to the first activity, which is modeled with a round-edged rectangle.
3. Now, connect each activity to other activities with lines that demonstrate the stepwise flow of the entire process.
4. You can also try using swimlanes to represent the objects that perform each activity.

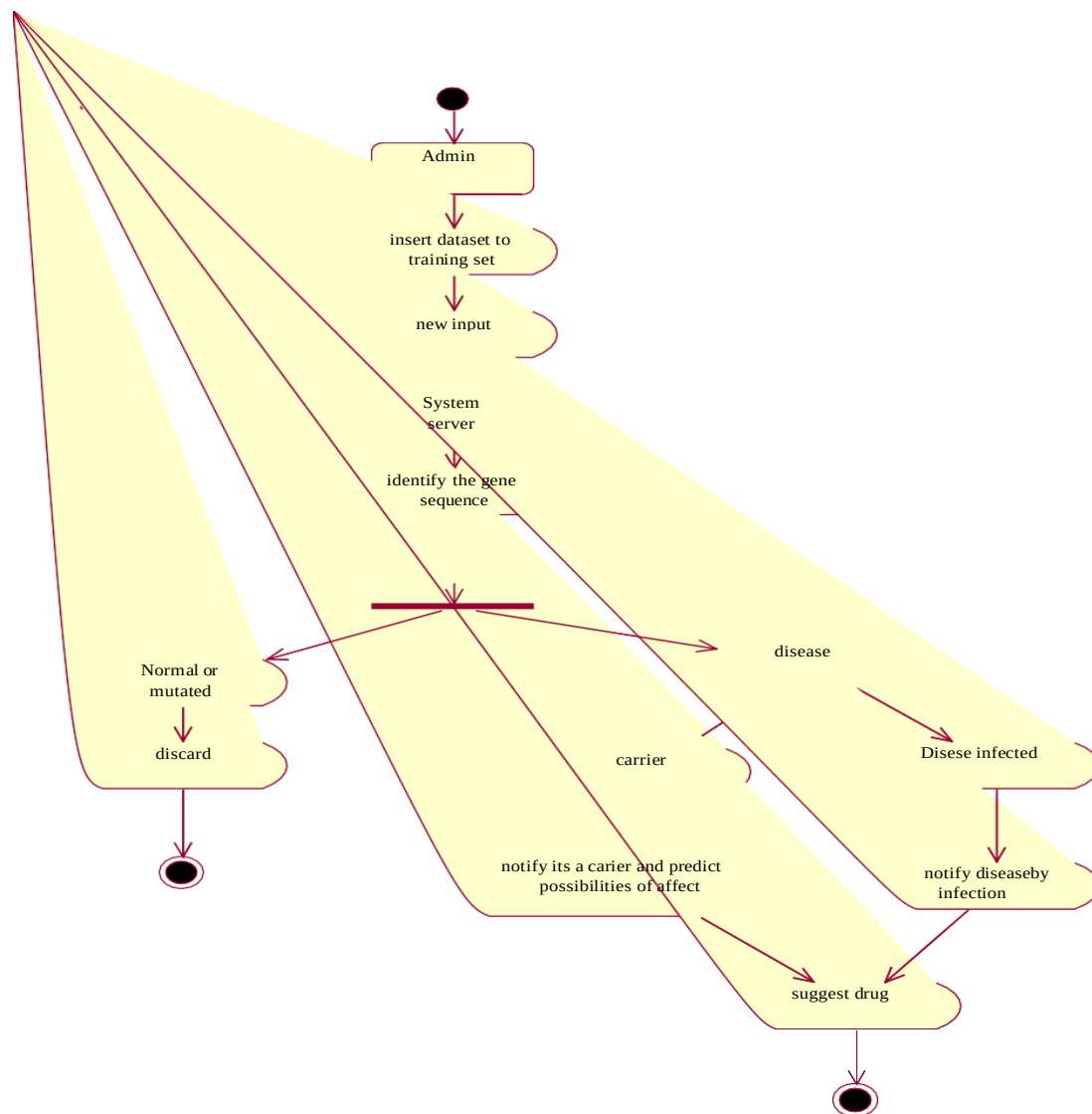


Fig. No: 3.5 Activity Diagram

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

The implementation of the project is to extract the Patient's gene sequence coding (Exons) from the given dataset and compare with the standardized Amino-acid sequence. The gene sequence is principally classified into two categories namely Introns & Exons. Introns are the Non-Coding sequence which is followed by Exons. Gene Sequence contains repeated of 4 Nucleotides namely Adenine (A), Thymine (T), Guanine (G), Cytosine (C). The proposed system is to recommend Diet pattern or any other natural drugs which can be recommended to those people who is expected to get into the disease in a course of time by verifying the mutated Genes. This paper infers that identifying disease based on gene using SVM algorithm. Through this system apart from disease identification we also suggest the drug based on the disease.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

4.1.1. Modules:

- A modular design reduces complexity, facilitates change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different parts of the system. Software with effective modularity is easier to develop because functions may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.
- Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comprehensibility, Modular Understandability, Modular continuity, Modular Protection.
- The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

- I. USER INTERFACE DESIGN
- II. DATA SET MAINTENANCE
- III. OUTLAYER REMOVAL
- IV. DISEASE PREDICTION
- V. DRUG SUGGESTION

I. User Interface Design:

User interface design, in this module User will be updating the Genome dataset of the Patient to the HDFS Server for Processing. To develop our application, we use NetBeans as IDE and MS SQL as a back end. All inputs and output will be put and get through this IDE only.

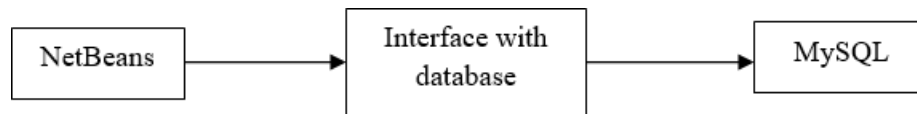


Fig. No: 4.1. User Interface Design Diagram

II. Data Set Maintenance:

The Server will monitor the entire Database. The database contains the Patient's Genome dataset along with Disease dataset for Comparison. Also the Server will store the entire information in their database. Also the Server has to establish the connection to communicate with the Users. Admin will add all the different types of gene datasets like parent gene, normal gene, mutated gene.



Fig. No: 4.2. Data Set Maintenance Diagram

III. Out layer Removal:

A gene mutation is a permanent alteration in the DNA sequence that makes up a gene, such that the sequence differs from what is found in most people. Mutations range in size; they can affect anywhere from a single DNA building block (base pair) to a large segment of a chromosome that includes multiple genes. The mutated gene is considered as equal to normal gene and it will be removed from the datasets.

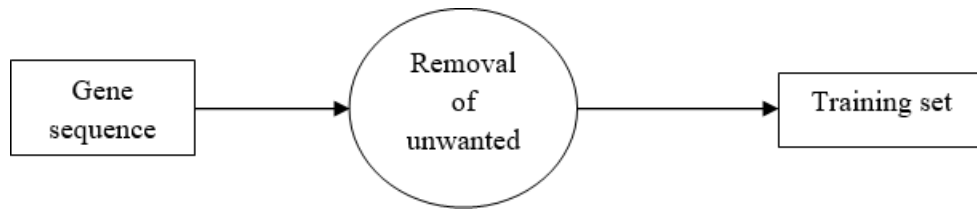


Fig. No: 4.3. Outer layer Removal Diagram

IV. Disease Prediction:

In this module we identify the disease by implementing SVM algorithm. we will identify the disease type that means, the gene which will given as a input, that will send to the trained dataset for comparison and identify the disease type whether it is gene based disease or infection based disease. if carrier-based disease found system will predict disease name.



Fig. No: 4.4. Disease Prediction Diagram

V. Drug Suggestion:

In this module after disease detection system will notify the possibilities of disease. And system will suggest the drug to that infected person. we suggest both English and herbal medicine as per user's requirement.

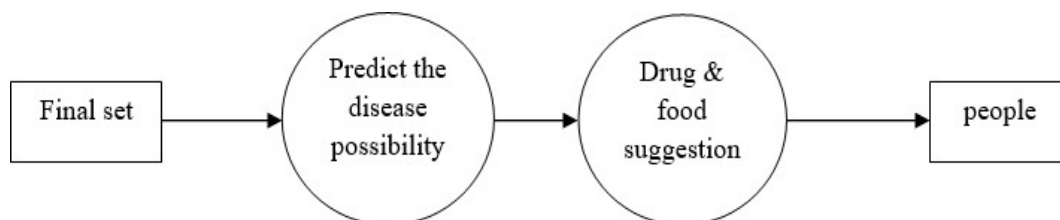


Fig. No: 4.5. Drug Suggestion Diagram

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

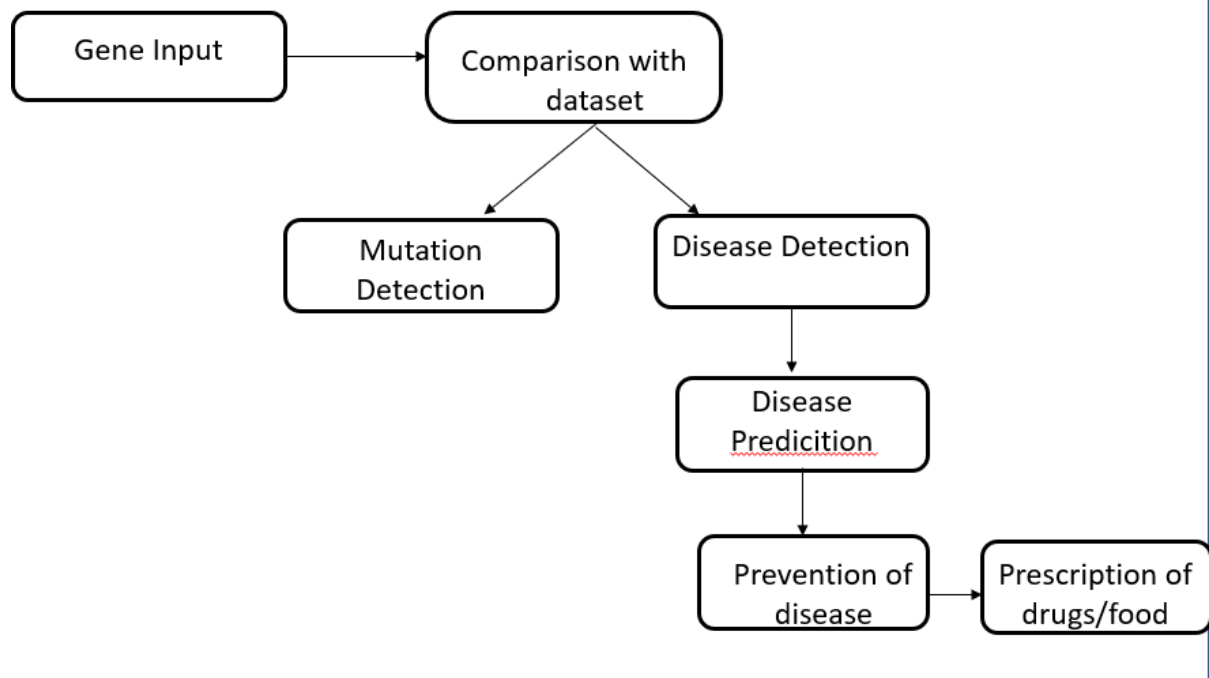


Fig 4.6: System Architecture

The overall architecture describes about user will upload gene sequence, after that those sequence will analyze using SVM algorithm in hadoop analyzer. Hadoop analyzer will translate the gene into short form of every gene sequence. After the translation it will form like a huge amount short form format which will predict the disease type and drugs and food suggestion to user.

4.3 Data Flow Diagram:

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships. Data flow diagrams visually represent systems and processes that would be hard to describe in a chunk of text. You can use these diagrams to map out an existing system and make it better or to plan out a new system for implementation. Visualizing each element makes it easy to identify inefficiencies and produce the best possible system.

I. Level 0:

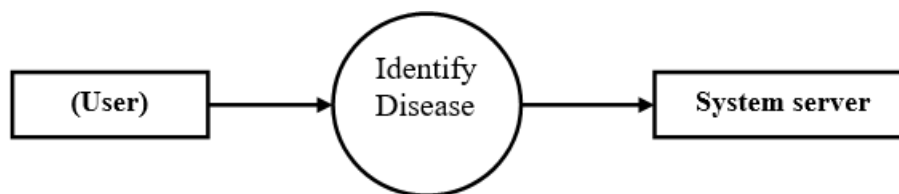


Fig 4.7: System Design Level 0

II. Level 1:

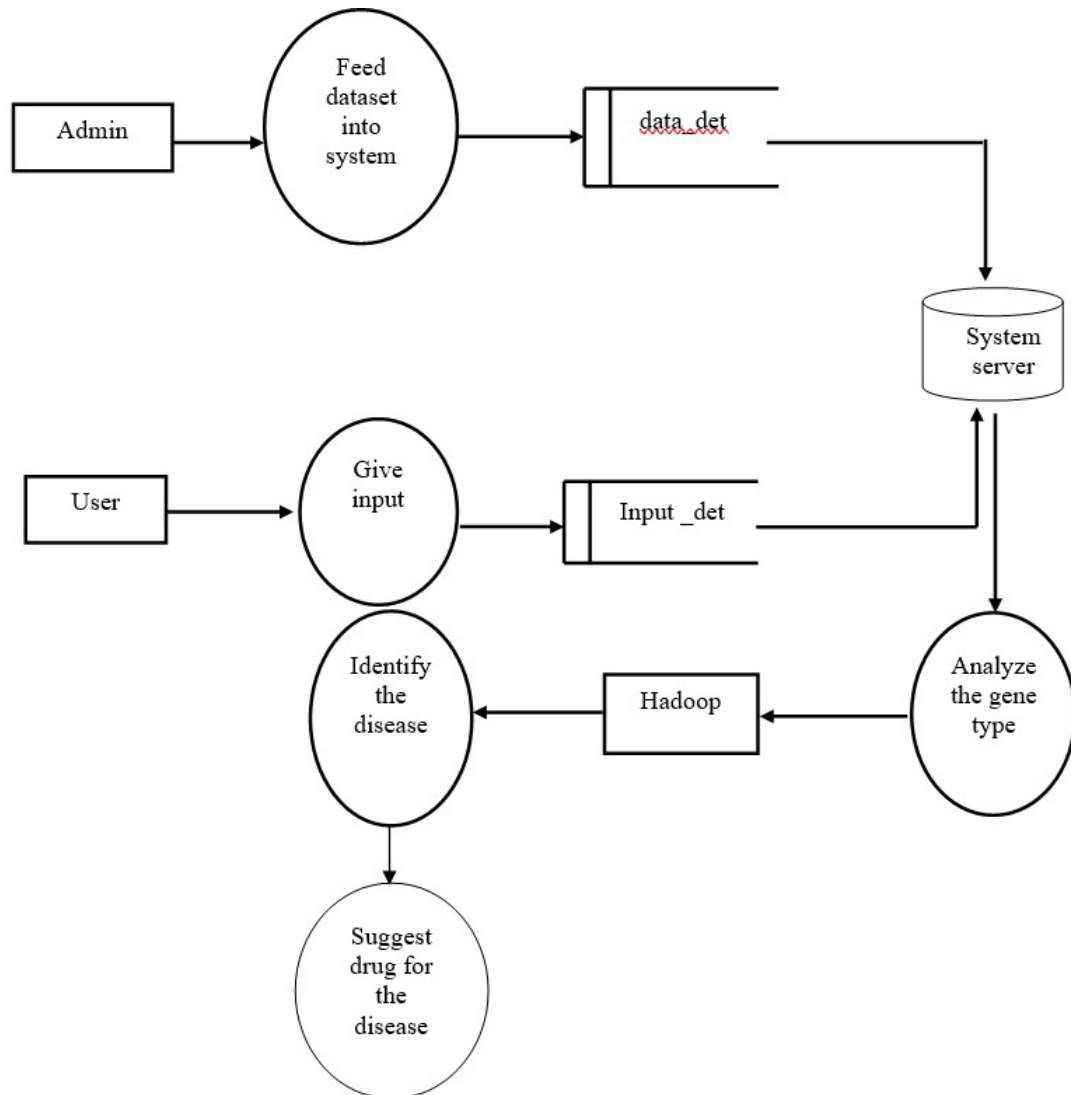


Fig 4.8: System Design Level 1

4.4 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

Verification of genetic sequences in real time is an area of study that is making significant headway and has the potential to bring about a sea change in the way genetic diseases are identified and treated. Scientists are now able to swiftly and precisely check genetic sequences in by applying cutting-edge technologies such as next-generation sequencing and bioinformatics. This is really useful for figuring out how to treat illnesses.

The mapping of illnesses is one of the most important applications of genomic sequence verification. During this stage of the procedure, we search for the precise genetic mutations that are to blame for a disease. Scientists are able to produce targeted treatments and therapies that are suited to the individual genetic makeup of a patient because they have a greater grasp of the underlying genetic causes of an illness. Customized medicine refers to this approach, which has the ability to greatly improve the effectiveness of medicines for a wide range of hereditary diseases.

The investigation of genetic polymorphisms is one more significant application that makes significant use of genetic sequence verification. Polymorphisms are naturally occurring variants in a population's genetic code. These variants are called polymorphisms. These changes may have a substantial influence on an individual's risk of developing certain illnesses as well as their overall health. Scientists can better understand the genetic elements that contribute to the development of certain illnesses and potentially create novel treatments and therapies if they discover and research genetic polymorphisms in real time. This is made possible by detecting and analyzing genetic polymorphisms.

The use of disease mapping and polymorphism analysis in conjunction with genetic sequence verification has the potential to significantly advance our knowledge of genetic illnesses and the creation of novel therapies. In this piece, we will discuss the present status of technology as well as its applications in the detection and

treatment of diseases.

There has been significant progress made in genotyping technology in the last several years as well as an increase in the availability of these technologies. These technologies make it possible to sequence DNA in a quick and reliable manner, which supplies information on the genetic make-up of an individual in great detail. This information is essential for determining genetic predispositions to a variety of illnesses, as well as for the diagnosis and treatment of genetic disorders.

Both the speed and accuracy of genetic sequencing have seen significant improvements as a result of the use of next-generation sequencing (NGS) technology. In a very short amount of time, NGS technologies enable the simultaneous sequencing of millions of DNA fragments, which provides a full snapshot of an individual's genetic make-up. Because of this, our capability to detect genetic mutations and changes that are connected with certain illnesses has significantly increased.

Disease mapping is one of the most important uses of next-generation sequencing technology. The process of disease mapping entails identifying the exact genetic abnormalities that are to blame for a certain illness. Scientists are able to produce targeted treatments and therapies that are suited to the individual genetic makeup of a patient because they have a greater grasp of the underlying genetic causes of an illness. This method is known as customised medicine, and treatments for a wide range of hereditary diseases may benefit greatly from this.

Research into genetic polymorphisms is another another significant use of next-generation sequencing technology. Polymorphisms are naturally occurring variants in a population's genetic code. Scientists can better understand the genetic elements that contribute to the development of certain illnesses and potentially create novel treatments and therapies if they discover and research.

4.4.1 Java

The Java platform is the ideal platform for network computing. Running across all platforms -- from servers to cell phones to smart cards -- Java technology unifies business infrastructure to create a seamless, secure, networked platform for your business. The Java platform benefits from a massive community of developers and supporters that actively work on delivering Java technology-based products and services as well as evolving the platform through an open, community-based, standards organization known as the Java Community Process program. You can find Java technology in cell phones, on laptop computers, on the Web, and even trackside at Formula One Grand Prix races. The fact is today; you can find Java technology just about everywhere!

I. Business Benefits

- A richer user experience - Whether you're using a Java technology-enabled mobile phone to play a game or to access your company's network, the Java platform provides the foundation for true mobility. The unique blend of mobility and security in Java technology makes it the ideal development and deployment vehicle for mobile and wireless solutions.
- The ideal execution environment for Web services - The Java and XML languages are the two most extensible and widely accepted computing languages on the planet, providing maximum reach to everyone, everywhere, every time, to every device and platform.
- Enabling business from end to end - Java offers a single, unifying programming model that can connect all elements of a business infrastructure.

II. Features of Java

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. Java is considered by many as one of the most influential programming languages of the 20th century, and widely used from application software to web application. The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of their Java technologies under the GNU General Public License.

III. Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- ☐ Simple
- ☐ Architecture neutral
- ☐ Object oriented
- ☐ Portal
- ☐ Distributed
- ☐ High performance

Each of the preceding buzzwords is explained in *The Java Language Environment*, a white paper written by James Gosling and Henry Chilton. In the Java programming language, all source code is first written in plain text files ending with the .java extension.

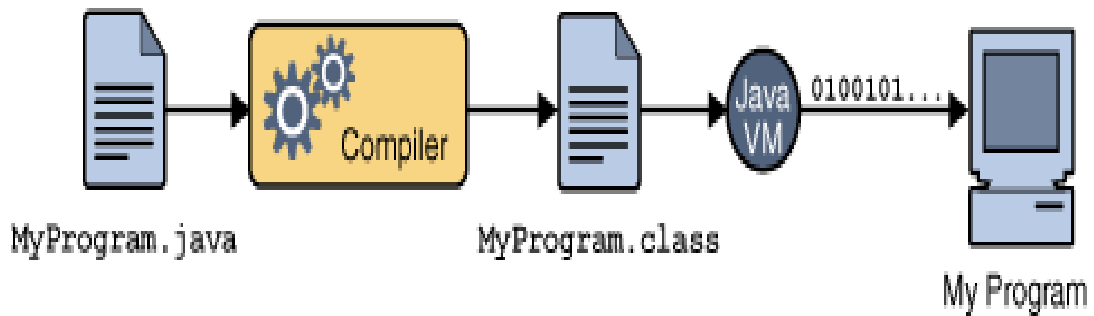


Fig 4.9 Software Development Process

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris Operating System (Solaris OS), Linux, or Mac OS. Some virtual machines, such as the Java Hotspot virtual machine, perform additional steps at runtime to give your application a performance boost. This includes various tasks such as finding performance bottlenecks and recompiling (to native code) frequently used sections of code.

IV. Java Platform

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Microsoft Windows, Linux, Solaris OS, and Mac OS. Most platforms can be described as a combination of the operating system and underlying hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- ☐ The *Java Virtual Machine*
- ☐ The *Java Application Programming Interface (API)*

V. Java Virtual Machine

A Java virtual machine (JVM) is a virtual machine that can execute Java byte code. It is the code execution component of the Java software platform. A Java virtual machine is a program which executes certain other programs, namely those containing Java byte code instructions. JVMs are most often implemented to run on an existing operating system, but can also be implemented to run directly on hardware. A JVM provides an environment in which Java byte code can be executed, enabling such features as automated exception handling, which provides root-cause debugging information for every software error (exception), independent of the source code.

A JVM is distributed along with a set of standard class libraries that implement the Java application programming interface (API). These libraries, bundled together with the JVM, form the Java Runtime Environment (JRE). JVMs are available for many hardware and software platforms. The use of the same byte code for all JVMs on all platforms allows Java to be described as a write once, run anywhere programming language, versus write once, compile anywhere, which describes cross- platform compiled languages. Thus, the JVM is a crucial component of the Java platform.

VI. Java API

The API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as *packages*.

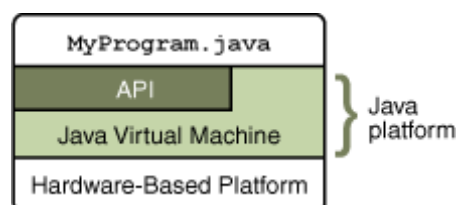


Fig 4.10 API and JV

VII. Java Runtime Environment

The Java Runtime Environment, or JRE, is the software required to run any application deployed on the Java Platform. End-users commonly use a JRE in software packages and Web browser plug-in. Sun also distributes a superset of the JRE called the Java 2 SDK (more commonly known as the JDK), which includes development tools such as the Java compiler, Javadoc, Jar and debugger.

One of the unique advantages of the concept of a runtime engine is that errors (exceptions) should not 'crash' the system. Moreover, in runtime engine environments such as Java there exist tools that attach to the runtime engine and every time that an exception of interest occurs they record debugging information that existed in memory at the time the exception was thrown (stack and heap values).

VIII. Java server Page

Java Server Pages (JSPs) are server-side Java EE components that generate responses, typically HTML pages, to HTTP requests from clients. JSPs embed Java code in an HTML page by using the special delimiters `<%` and `%>`. A JSP is compiled to a Java *servlet*, a Java application in its own right, the first time it is accessed. After that, the generated servlet creates the response.

IX. Swing Application

Swing is a graphical user interface library for the Java SE platform. It is possible to specify a different look and feel through the pluggable look and feel system of Swing. Clones of Windows, GTK+ and Motif are supplied by Sun. Apple also provides an Aqua look and feel for Mac OS X. Where prior implementations

Of these looks and feels may have been considered lacking, Swing in Java SE 6 addresses this problem by using more native GUI widget drawing routines of the underlying platforms.

This example Swing application creates a single window with "Hello, world!" inside:

```
// Hello.java (Java SE
5)
import javax.swing.*;
public class Hello extends
JFrame { PublicHello () {
setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
add(new JLabel("Hello, world!"));
pack();

} public static void main(String[]
args) { new Hello().setVisible(true);
}}
```

The first import includes all of the public classes and interfaces from the javax.swing package. The Hello class extends the Frame class; the Frame class implements a window with a title bar and a close control. The Hello () constructor initializes the frame by first calling the super class constructor, passing the parameter "hello", which is used as the window's title. It then calls the setDefaultCloseOperation (into) method inherited from Frame to set the default operation when the close control on the title bar is selected to Window Constants.EXIT ON CLOSE this causes the J Frame to be disposed of when the frame is closed (as opposed to merely hidden), which allows the JVM to exit and the program to terminate. inherited from the Component super class with the Boolean parameter true. Once the frame is displayed, exiting the main method does not cause the program to terminate because the AWT event dispatching thread remains active until all of the Swing top- level windows have been disposed.

X. Java Platform, Standard Edition

Java Platform, Standard Edition or Java SE is a widely used platform for programming in the Java language. It is the Java Platform used to deploy portable applications for general use. In practical terms, Java SE consists of a virtual machine, which must be used to run Java programs, together with a set of libraries (or "packages") needed to Allow the use of file systems, networks, graphical interfaces, and so on, from within those programs. Some of the general purpose packages in J2SE are as follows

- ☐ Java.lang
- ☐ Java.io
- ☐ Java.math
- ☐ Java.net
- ☐ Java.util

XI. Introduction To Apache Tom Cat

Tomcat, developed by Apache (www.apache.org), is a standard reference implementation for Java servlets and JSP. It can be used standalone as a Web server or be plugged into a Web

Server like Apache, Netscape Enterprise Server, or Microsoft Internet Information Server. There are many versions of Tomcat. This tutorial uses Tomcat 5.5.9 as an example. The tutorial should also apply to all later versions of Tomcat. To start Tomcat, you have to first set the JAVA_HOME environment variable to the JDK home directory using the following command. The JDK home directory is where your JDK is stored. On my computer, it is c:\Program Files\jdk1.6.0_24. The apache tomcat server is an open source, java-based web application container that was created to servlets and java server page (JSP) web applications. It was created under the apache-Jakarta subproject; however, due to its popularity, it is now hosted as separate apache project, where it is supported and enhanced by a group of volunteers from the open source java community.

The Tomcat manager web application is packaged with the tomcat server. It is installed in the context path of manager and provides the basic functionality to manage web applications running in the tomcat server from any web browser. Some of the provided functionality includes the ability to install, start, stop, remove, and report on web applications. Covers the details of the tomat manager web application.

A. The Server

The first container element referenced in this snippet is the <Server> element. It represents the entire Catalina servlet engine and is used as a top-level element for a single Tomcat instance. The <Server> element may contain one or more <Service> containers.

B. The Service

The next container element is the <Service> element, which holds a collection of one or more <Connector> elements that share a single <Engine> element. N- number of <Service> elements may be nested inside a single <Server> element.

C. The Connector

The next type of element is the <Connector> element, which defines the class that does the actual Handling requests and responses to and from a calling clientapplication.

D. The Engine

The third container element is the <Engine> element. Each defined <Service> can have only one <Engine> element and this single <Engine> component handles all requests received by all of the defined <Connector> components defined by a parent service.

E. The Host

The <Host> element defines the virtual hosts that are contained in each instance of a Catalina <Engine>. Each <Host> can be a parent to one or more web applications, with each being represented by a <Context> component.

F. The Context

The <Context> element is the most commonly used container in a Tomcat instance. Each <Context> element represents an individual web application that is running within a defined <Host>. There is no limit to the number of contexts that can be defined within a <Host>.

XII. Java Web Applications

The main function of the Tomcat server is to act as a container for Java web applications. Therefore, before we can begin our Tomcat-specific discussion, a brief introduction as to exactly what web applications are is in order. The concept of a web application was introduced with the release of the Java servlets specification 2.2. According to this specification, “a web application is a collection of servlets, html pages, classes, and other resources that can be bundled and run on multiple containers from multiple vendors.” What this really means is that a web application is a container that can hold any combination of the following list of objects:

- ☐ Servlets
- ☐ Java Server Pages (JSPs)
- ☐ Utility classes
- ☐ Static documents, including HTML, images, JavaScript libraries, cascading style sheets, and so on
- ☐ Client-side classes

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

5.1.1 Big Data

Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications. The challenges include analysis, capture, duration, search, sharing, storage, transfer, visualization, and privacy violations. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, prevent diseases, combat crime and so on. So we can implement big data in our project because every employ has instructed information so we can make analysis on this data.

I. Characteristics of Big Data

Big data is a term utilized to refer to the increase in the volume of data that are difficult to store, process, and analyze through traditional database technologies. The term "big data" is relatively new in IT and business. However, several researchers and practitioners have utilized the term in previous literature. For instance, referred to big data as a large volume of scientific data for visualization. Several definitions of big data currently exist." Meanwhile and defined big data as characterized by three Vs: volume, variety, and velocity. The terms volume, variety, and velocity were originally introduced by Gartner to describe the elements of big data challenges. IDC also defined big data technologies as "a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling the high velocity capture, discovery, and/or analysis

5.1.2 Java

The Java platform is the ideal platform for network computing. Running across all platforms -- from servers to cell phones to smart cards -- Java technology unifies business infrastructure to create a seamless, secure, networked platform for your business. The Java platform benefits from a massive community of developers and supporters that actively work on delivering Java technology-based products and services as well as evolving the platform through an open, community-based, standards organization known as the Java Community Process program. You can find Java technology in cell phones, on laptop computers, on the Web, and even trackside at Formula One Grand Prix races. The fact is today; you can find Java technology just about everywhere.

5.1.3 Mysql

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site (<http://www.mysql.com/>) provides the latest information about MySQL software. A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

5.2 ALGORITHMS

Support Vector Machines (SVMs) are a popular machine learning algorithm that can be used for genetic sequence verification with disease mapping and polymorphism. SVMs are based on the principle of finding a hyperplane in a high-dimensional space that can best separate different classes of data. In the context of genetic data, SVMs can be used to classify different genetic sequences as being associated with a particular disease or polymorphism.

SVMs are particularly effective when dealing with high-dimensional datasets, where the number of features (e.g., genetic variants) is much larger than the number of samples. SVMs work by mapping the input data into a higher-dimensional space, where a hyperplane can be found that best separates the different classes. SVMs can handle non-linear relationships between input features using kernel functions that transform the data into a new space.

When using SVMs for genetic sequence verification with disease mapping and polymorphism, the SVM model is trained using a set of labeled genetic data. This labeled data is used to find the best hyperplane that separates the data into different classes (e.g., disease versus healthy). Once the model is trained, it can be used to predict the class of new genetic data. One of the advantages of SVMs is their ability to handle noise and outliers in the data. SVMs can also be used with different kernel functions to handle different types of data and non-linear relationships. However, SVMs can be computationally expensive and require significant computational resources, especially when dealing with large datasets. The selection of the kernel function and other hyper parameters of the SVM model can also impact its performance, and therefore, require careful tuning.

Overall, SVMs can be a useful algorithm for genetic sequence verification with disease mapping and polymorphism. However, like any machine learning algorithm, their effectiveness depends on the quality of the data used to train and test the model, and the proper selection and tuning of its parameters.

5.3 SYSTEM TESTING

Generally testing process is carried out to ensure that the system has been developed according to the required specifications and the expected output is properly obtained. There are two main categories of testing namely, the White Box Testing and the Black Box Testing. Each of this testing in turn consists of many types of testing.

I. Taxonomy of Testing

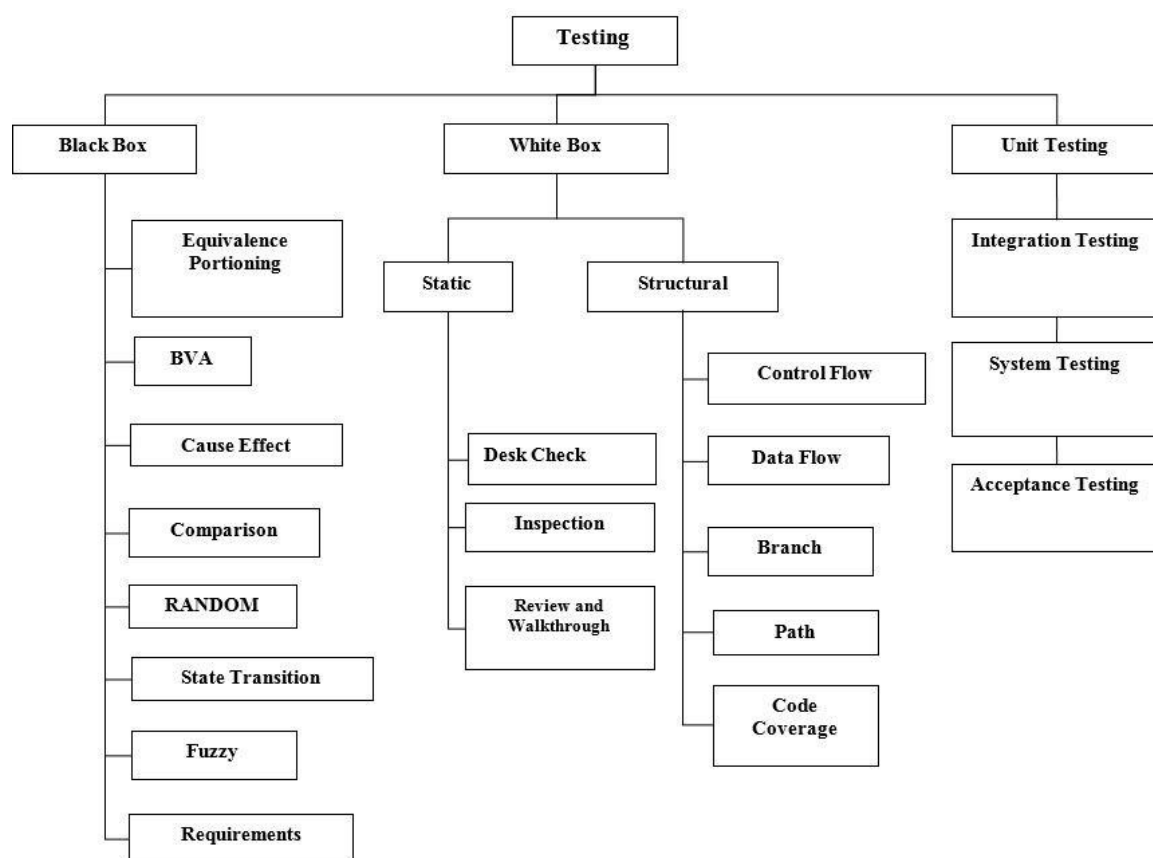


Fig 5.1 Taxonomy of Testing

II. White Box Testing

White Box testing is also known as glass box testing. This type of testing, tests the internal structure of the program. This can be applied at the unit, integration and system levels of testing. Mostly, it is used in the unit level of the software testing process. Sometimes it may not reveal defects in areas which have not been implemented.

III. Black Box Testing

It is amongst the two methods of mostly used testing methods. This tests the main functionality of the program. It can be applied to every level of testing such as Unit, Integration, System and Acceptance levels of testing. Exhaustive input testing is required to find all errors. For doing this type of testing knowing the internal code and how it works is not needed but what it is supposed to do is known by the person who is performing the test. The test cases are developed based on the specific requirements according to the goals. There are Boundary Value Analysis, Class Partitioning, Cause Effect Graph etc.,

IV. Unit Testing:

Unit testing is also known as Module Testing which focuses on verification efforts on the module. The module is tested separately and this is carried out at the programming stage itself. Unit test comprises of the set of tests performed by an individual programmer before integration of the unit into the system. This will help to test each and every single part or we can say as each and every module completely

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 RESULTS:

The suggested method for genetic sequence verification is intended to swiftly and precisely identify genetic mutations and variants linked to hereditary illnesses. DNA is isolated, fragmented, ligated using an adaptor, sequenced, and data is analyzed using methods like Linear SVM and Multi Class SVM. The Support Vector Machine approach was determined to be the most dependable algorithm for this purpose.

A wealth of information is available for disease diagnosis and therapy thanks to the technology's usage in genetic polymorphism analysis and disease mapping. Genetic information is compared to a database of known genetic mutations and variations associated with various diseases and traits. This makes it possible to identify people who are at a high risk of contracting a specific disease and to suggest specific treatment strategies based on the findings of genetic verification.

6.2 DISCUSSIONS

Precision medicine and personalized medicine could both be revolutionized by real-time DNA sequence verification. The development of focused treatment regimens and improved patient outcomes are made possible by the capacity to quickly and precisely detect genetic mutations and variants linked to genetic illnesses.

However, the suggested approach only considers genetic mutations and variations linked to certain conditions or traits, although other elements including environmental and lifestyle factors also play a role in the emergence of genetic disorders. Consequently, for personalised treatment and precision healthcare, a thorough strategy that considers both genetic and non-genetic elements may be required.

In conclusion, genetic sequence verification has a bright future for precision healthcare and tailored therapy. A lot more study is required to fully grasp the potential of the suggested process of DNA separation, fragmentation, adaptor ligation, sequencing, and data analysis using algorithms like Linear SVM and Multi Class SVM.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

Here we can identify disease based on gene using SVM algorithm. We can also find out whether it is polymorphism. Through this system apart from disease identification we also suggest the drug based on the disease. Genetic sequence verification with disease mapping and polymorphism using SVM is a promising approach for detecting and predicting genetic variations associated with various diseases. The use of Support Vector Machines (SVM) for classification and prediction tasks has proven to be effective in analyzing large datasets of genetic information. With this approach, genetic sequences can be analyzed, allowing for quicker diagnosis and treatment of genetic diseases. This method also enables the identification of polymorphisms, which can provide insights into genetic diversity and evolution. However, further research is needed to improve the accuracy and efficiency of this approach, and to explore its potential applications in various fields such as personalized medicine, agriculture, and environmental science.

7.2 FUTURE WORK

There are several potential areas of future work for Disease Prediction Using Genetic Mutation Through Big Data Analysis using SVM:

- i. **Improvement of SVM algorithm:** There is always room for improvement in the SVM algorithm used for genetic sequence verification. Future research could explore new approaches to SVM optimization, such as tuning of hyper parameters
- ii. **Integration with other Machine Learning Algorithms:** There is also potential for integrating SVM with other machine learning algorithms for sequence verification. The combination of SVM with deep learning algorithms like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) could improve the accuracy of sequence verification even further.
- iii. **Expansion of Disease Mapping:** Disease mapping is a crucial aspect of genetic sequence verification, and it would be beneficial to expand the scope of disease mapping to include more genetic diseases. This would require more extensive databases of genetic variants and associated diseases.
- iv. **Real-time Monitoring of Sequencing:** Real-time monitoring of genetic sequencing is another area of potential future work. By integrating real-time monitoring with SVM-based sequence verification, we can ensure that sequencing errors are detected and corrected as soon as possible.
- v. **Integration with Cloud Computing:** Cloud computing has revolutionized many areas of science and technology, including genetic sequencing. Future research could explore the potential of integrating SVM-based sequence verification with cloud-based computing resources, making it more accessible to researchers and clinicians around the world.

7.3 IMPLEMENTATION ISSUES

- i. Hardware requirements:** genetic sequence verification with disease mapping and polymorphism using SVM requires significant computational resources. Adequate hardware infrastructure, such as high-performance servers, GPUs, or cloud-based computing platforms, may be required to implement the solution.
- ii. Software development:** Developing software to implement real-time genetic sequence verification with disease mapping and polymorphism using SVM requires expertise in machine learning and software development. It may also require familiarity with specific programming languages, such as Python or R.
- iii. Data acquisition and pre-processing:** Obtaining and processing large amounts of genetic data requires specialized knowledge and tools. Data cleaning, normalization, and feature extraction techniques may need to be applied to the raw data before it can be used to train and validate the SVM model.
- iv. Integration with existing systems:** The SVM model must be integrated with existing clinical or laboratory systems for real-time disease mapping and prediction. This can be a complex process that requires careful consideration of system architecture, data exchange protocols, and security measures.
- v. Maintenance and updates:** Once the SVM model is deployed, it will require ongoing maintenance and updates. This includes monitoring its performance, updating the model with new data, and retraining the model as needed.
- vi. Regulatory and compliance requirements:** The use of genetic data in healthcare applications is subject to regulatory and compliance requirements, such as HIPAA in the United States. Compliance with these requirements must be ensured when implementing real-time genetic sequence verification with disease mapping and polymorphism using SVM.

CHAPTER 8

REFERENCES

- [1] Cheng, X., Chen, X., Zhu, Y., Li, F., Wang, J., Wang, X., & Wu, F. X. (2020). Polymorphism-based prediction of genetic diseases using support vector machine. *Frontiers in Genetics*, 11, 842.
- [2] Lu, Y., Li, Y., Zhou, Y., & Chen, Y. P. (2019). Genomic polymorphism-based analysis and prediction of disease risks using support vector machines. *BMC Bioinformatics*, 20(1), 60.
- [3] Zhang, H., Li, Y., Zhou, Y., & Chen, Y. P. (2018). Genomic polymorphism-based analysis and prediction of disease risks using support vector machines. *PloS one*, 13(11), e0207725.
- [4] Lin, Y. K., & Chang, K. M. (2018). Machine learning in clinical genetic diagnosis of complex diseases. *Genes*, 9(9), 438.
- [5] Lee, J., Lee, S., Kim, S., & Lee, K. (2015). Support vector machine-based prediction of genetic interactions using functional genomics data. *BMC Bioinformatics*, 16(1), 196.
- [6] Anand, A. and Shakti, D., 2015. Prediction of diabetes based on personal lifestyle indicators. In *Next generation computing technologies (NGCT)*, 2015 1st international conference on (pp. 673–676). IEEE.
- [7] Cho, Y. S., Park, Y. R., & Seo, J. (2014). A hybrid approach using support vector machine and logistic regression for disease prediction. *Journal of medical systems*, 38(6), 52.
- [8] Pattekari, S.A. and Parveen, A., 2012. Prediction system for heart disease using Naïve Bayes. *International Journal of Advanced Computer and Mathematical Sciences*, 3(3), pp. 290–294.
- [9] Sharma, M. and Majumdar, P.K., 2009. Occupational lifestyle diseases: An emerging issue. *Indian Journal of Occupational and Environmental medicine*, 13(3), pp. 109–112.
- [10] Kanchan, B.D. and Kishor, M.M., 2016. Study of machine learning algorithms for special disease prediction using principal of component analysis. In *Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICCC)*, 2016 International Conference on (pp. 5–10). IEEE.

- [11] Kazeminejad, A., Golbabaei, S. and Soltanian-Zadeh, H., 2017. Graph theoretical metrics and machine learning for diagnosis of Parkinson's disease using rs-fMRI. In Artificial Intelligence and Signal Processing Conference (AISP), (pp. 134– 139). IEEE.
- [12] Milgram, J., Cheriet, M. and Sabourin, R., 2006. "One against one" or "one against all": Which one is better for handwriting recognition with SVMs?. Tenth International Workshop on Frontiers in Handwriting Recognition, La Baule (France), Suvisoft, 2006.
- [13] Hossain, R., Mahmud, S.H., Hossin, M.A., Noori, S.R.H. and Jahan, H., 2018. PRMT: Predicting Risk Factor of Obesity among MiddleAged People Using Data Mining Techniques. *Procedia Computer Science*, 132, pp. 1068–1076.
- [14] Sayali Ambekar and Dr.Rashmi Phalnikar, 2018. Disease prediction by using machine learning, *International Journal of Computer Engineering and Applications*, vol. 12, pp. 1–6.
- [15] Mishra, A.K., Keserwani, P.K., Samaddar, S.G., Lamichaney, H.B. and Mishra, A.K., 2018. A decision support system in healthcare prediction. In *Advanced Computational and Communication Paradigms* (pp. 156–167). Springer, Singapore.
- [16] Explaining the basics of machine learning, algorithms and applications, Available[Online],<https://www.hackerearth.com/blog/machinelearning/explainingbasics-machine-learning-algorithms-applications/> [Accessed on June 27, 2018]
- [17] Suzuki, A., Lindor, K., St Saver, J., Lymp, J., Mendes, F., Muto, A., Okada, T. and Angulo, P., 2005. Effect of changes on body weight and lifestyle in nonalcoholic fatty liver disease. *Journal of Hepatology*, 43(6), pp. 1060–1066.
- [18] K. M. Boycott, M. R. Vanstone, D. E. Bulman and A. E. MacEnzie, "Rare-disease genetics in the era of next-generation sequencing: discovery to translation", in *Nature Reviews Genetics*, vol. 14(10), pp. 681–691, 2013.
- [19] C. M. Condit, P. J. Achter, I. Lauer and E. Sefcovic, "The changing meanings of "mutation:" A contextualized study of public discourse", in *Human Mutation*, vol. 19(1), pp. 69–75, 2002.
- [20] W. Raghupathi and Viju Raghupathi. "Big Data Analytics in Healthcare: Promise and Potential.", in *Health Information Science and Systems*, 2:3, 2014.

APPENDIX

SOURCE CODE

I. Homepage:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed

    String uname=jTextField1.getText().toString();
    String
    pass=jPasswordField1.getText().toString();
    AddService _ad=new AddService();
    if(_ad.Login_Valid(uname, pass)){
        this.dispose();
        new HadoopUploadFile().setVisible(true);
    }else{
        JOptionPane.showMessageDialog(rootPane, "Invalid User Credentials");
    }

}

}

//GEN-LAST:event_jButton1ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton3ActionPerformed

    this.dispose();
    new Userregistration().setVisible(true);
}

//GEN-LAST:event_jButton3ActionPerformed

public static void main(String
args[]) {try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName()); break;
```

```

        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);

            } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);

        }

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new HomePage().setVisible(true);
            }
        }
    }
}

```

II. User Registration:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {GEN-
FIRST:event_jButton3ActionPerformed

String Name=jTextField1.getText().toString().trim(); String
UserName=jTextField2.getText().toString().trim(); String
Password=jTextField3.getText().toString().trim(); String
Email=jTextField4.getText().toString().trim(); String
ContactNo=jTextField5.getText().toString().trim(); String
Location=jTextField6.getText().toString().trim(); AddService
_ad=new AddService();
if(_ad.Add_User(Name, UserName, Password, Email, ContactNo, Location)>0){
    JOptionPane.showMessageDialog(rootPane, "UserDetails saved");
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
}else{
    JOptionPane.showMessageDialog(rootPane, "Connection-error");
}
}
}GEN-LAST:event_jButton3ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {GEN-
FIRST:event_jButton2ActionPerformed

// TODO add your handling code here:
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");
jTextField6.setText("");
}GEN-LAST:event_jButton2ActionPerformed
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {GEN-
FIRST:event_jButton4ActionPerformed

// TODO add your handling code here:
this.dispose();
```

```

        new HomePage().setVisible(true);
    } //GEN-LAST:event_jButton4ActionPerformed

    public static void main(String args[])
    { try {
        for
        (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels())
        {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Userregistration.class.getName()).log(java.util.log
        ging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(Userregistration.class.getName()).log(java.util.log
        ging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Userregistration.class.getName()).log(java.util.log
        ging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Userregistration.class.getName()).log(java.util.log
        ging.Level.SEVERE, null, ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() { new Userregistration().setVisible(true);
    }
}

```

III. Upload file:

```
private void upload_btActionPerformed(java.awt.event.ActionEvent evt){ //GEN-FIRST:event_upload_btActionPerformed try {

    // TODO add your handling code here:
    File f=new File(fileSelected.getAbsolutePath());
    File ff=new
File(constants.Constants.FILE_HADOOP_USER_IN_LOCATION);
    FileInputStream fin=new FileInputStream(f);
    String
readdata=""; int
i=0;
    while((i=fin.read())!=-1){
        byte b[]=new
        byte[fin.available()]; fin.read(b);

        readdata=new String(b);
    }
    fin.close();
    FileOutputStream fout=new FileOutputStream(ff);
    fout.write(readdata.getBytes());
    fout.close();
    // new HadoopAnalysis().processFiles(this.fileSelected);
    //JOptionPane.showMessageDialog(this, "Press ok to view file");
    //Desktop.getDesktop().open(new
File(constants.Constants.FILE_LOCATION));
    } catch (Exception ex) {
    Logger.getLogger(HadoopUploadFile.class.getName()).log(Level.SEVERE,
    null, ex);}
}
```

```

        finally{ fileSelected_TF.setText("");
        JOptionPane.showMessageDialog(rootPane, "File Upload Successfully");
    }

    }//GEN-LAST:event_upload_btActionPerformed
    public static void main(String args[])

        { try {

            for(javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {

                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;

                }

            }

        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(HadoopUploadFile.class.getName()).log(java.util.l
            ogging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(HadoopUploadFile.class.getName()).log(java.util.l
            ogging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(HadoopUploadFile.class.getName()).log(java.util.l
            ogging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(HadoopUploadFile.class.getName()).log(java.util.l
            ogging.Level.SEVERE, null, ex);

        }

        java.awt.EventQueue.invokeLater(new Runnable()

        { public void run() {

            new HadoopUploadFile().setVisible(true);

        }

        }

    }

```

SCREENSHOTS



A screenshot of a web application window titled "Home-Page". The background is a solid blue color. At the top center, the text "Home-Page" is displayed in a white, serif font. Below this, there are two white input fields. The first is labeled "UserName" in white text to its left. The second is labeled "Password" in white text to its left. Below the input fields, there are three buttons: "LOGIN" and "LOGOUT" are positioned side-by-side, and "REGISTER" is centered below them. All buttons have a light gray gradient and black text.

Home-Page

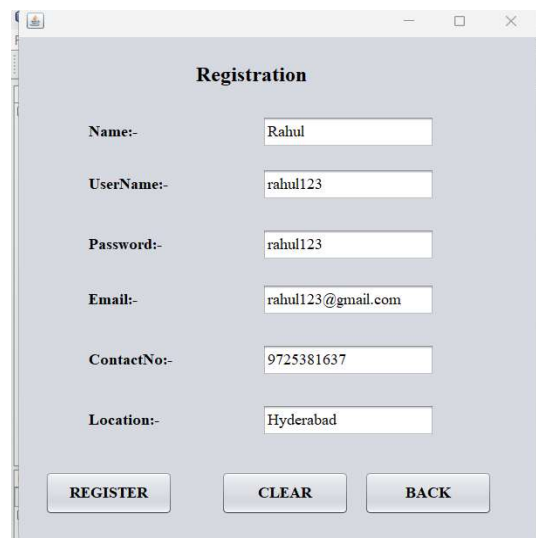
UserName

Password

LOGIN LOGOUT

REGISTER

LOGIN PAGE



A screenshot of a web application window titled "Registration". The background is a light gray color. At the top center, the text "Registration" is displayed in a bold, black, serif font. Below this, there are six rows of labels and input fields. The labels are "Name:-", "UserName:-", "Password:-", "Email:-", "ContactNo:-", and "Location:-", all in a black, serif font. The input fields contain the following text: "Rahul", "rahul123", "rahul123", "rahul123@gmail.com", "9725381637", and "Hyderabad". At the bottom, there are three buttons: "REGISTER", "CLEAR", and "BACK", all with a light gray gradient and black text.

Registration

Name:-

UserName:-

Password:-

Email:-

ContactNo:-

Location:-

REGISTER CLEAR BACK

Registration Page

Registration

Name:-

UserName:-

Password:-

Email:-

ContactNo:-

Location:-

Message: UserDetails saved

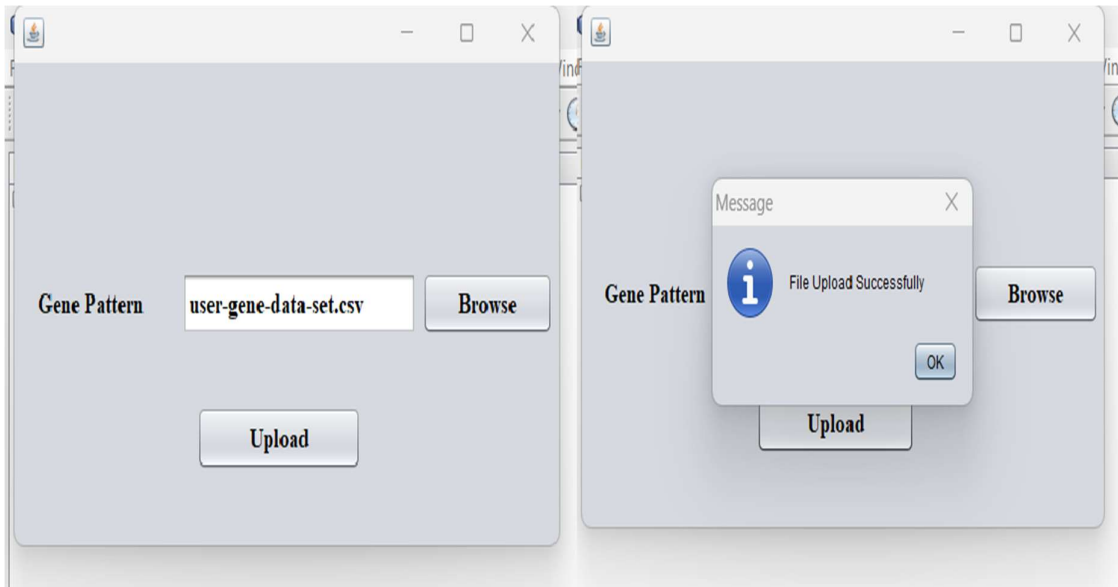
Registration of User

Home-Page

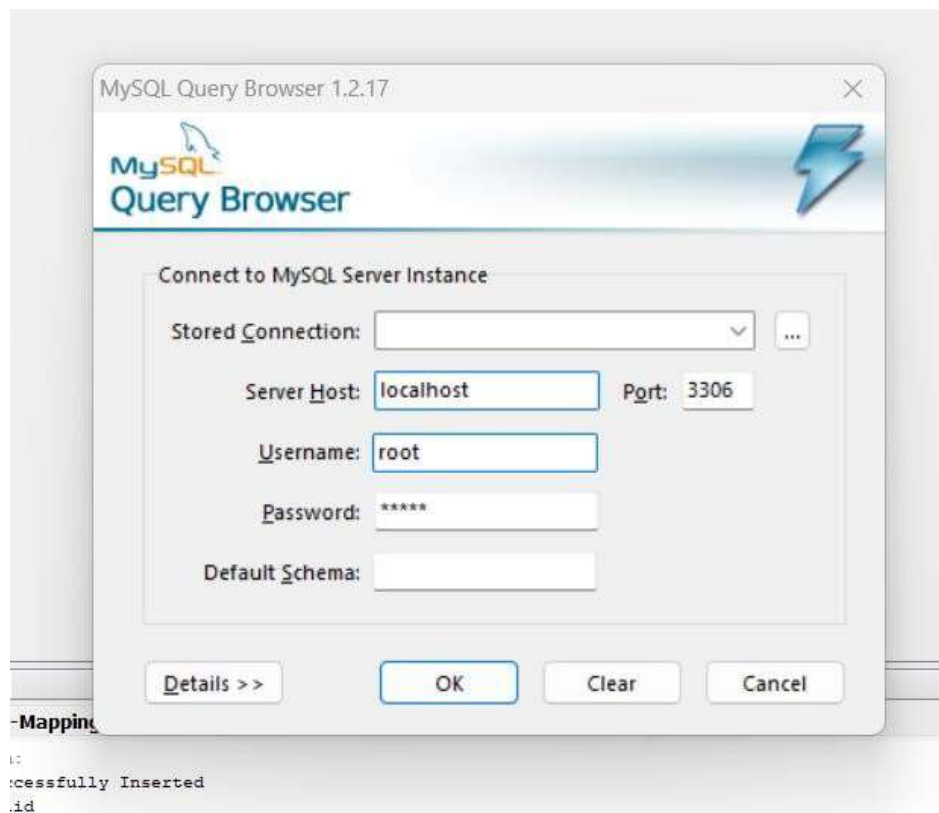
UserName

Password

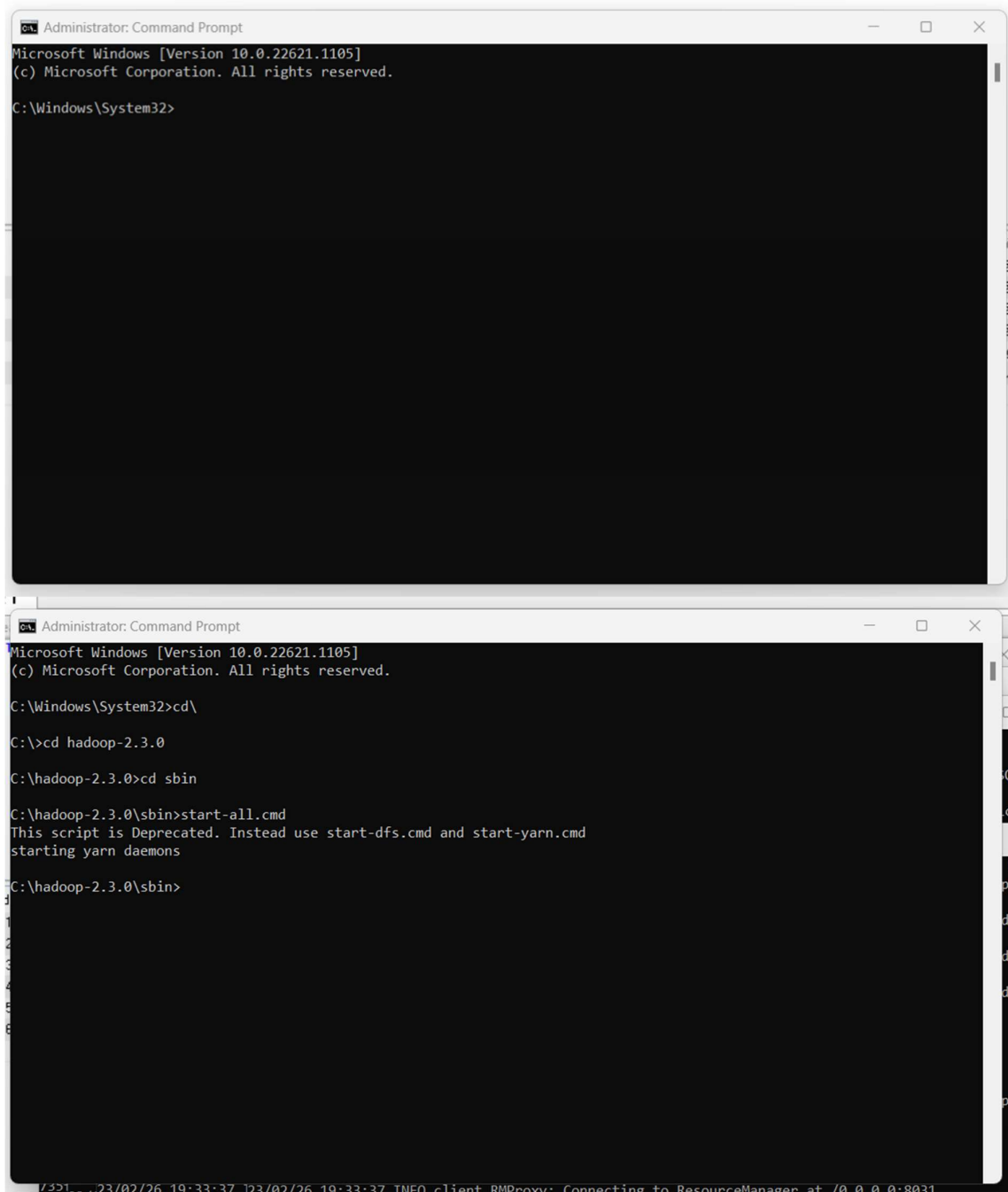
Login Of Registered User



Data-Set Upload



MySQL Login



The image displays two screenshots of a Windows Command Prompt window, titled "Administrator: Command Prompt".

The top screenshot shows the initial state of the command prompt with the following text:

```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>
```

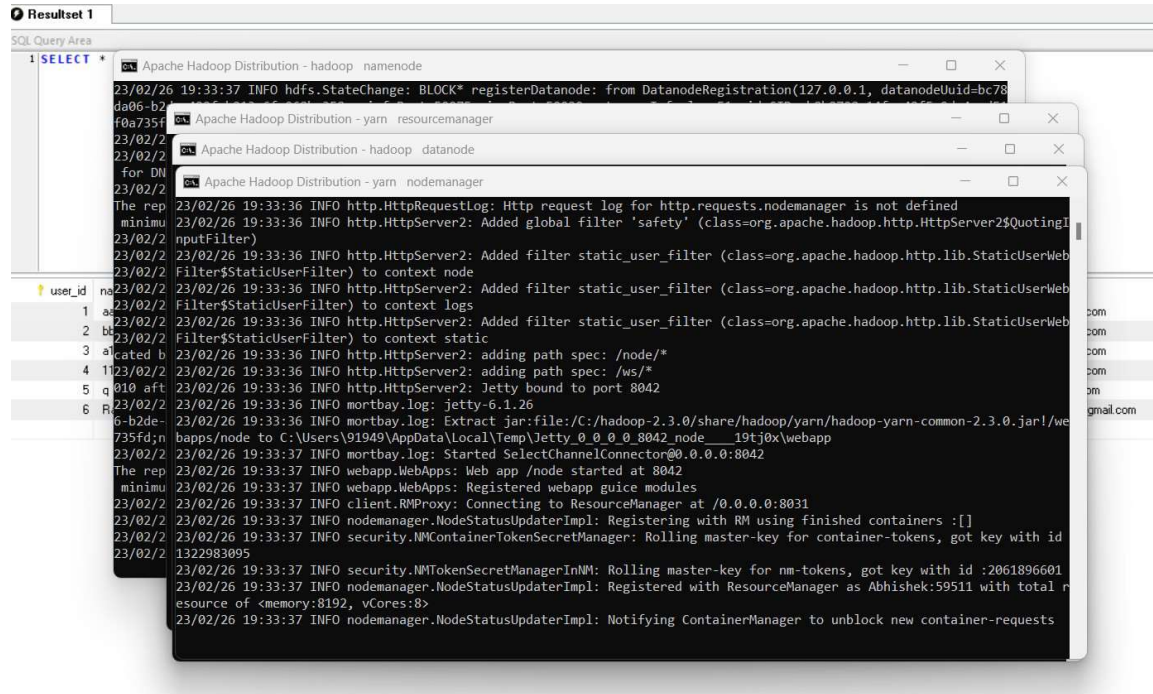
The bottom screenshot shows the same command prompt after several commands have been entered and executed:

```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd\
C:\>cd hadoop-2.3.0
C:\hadoop-2.3.0>cd sbin
C:\hadoop-2.3.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons
C:\hadoop-2.3.0\sbin>
```

At the bottom of the window, a status bar displays the following information: 1/321, 23/02/26 19:33:37, 23/02/26 19:33:37, INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8031.

Command Prompt



Hadoop Installisation



User Id Entry

C. RESEARCH PAPER

GENOMIC BIG DATA ANALYTICS FOR DISEASE PREDICTION: UNRAVELING GENETIC MUTATIONS TO ENHANCE PRECISION MEDICINE.

Vennapusa Shanmukendar Reddy
Dept. of Computer Science Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
shanmukendar906@gmail.com

Mrs. G. KAVITHA M.E, (PHD)
Assistant Professor
Sathyabama Institute of Science and
Technology
Chennai, India
gkavitha.cse@sathyabama.ac.in

Abstract— *The convergence of genomic big data analytics and precision medicine heralds a new era in healthcare, promising unprecedented insights into disease prediction and treatment. This article explores the transformative potential of genomic big data analytics in deciphering genetic mutations and their role in enhancing precision medicine strategies. Advancements in genomic sequencing technologies have led to the generation of vast repositories of genetic data, presenting both opportunities and challenges for disease prediction. By leveraging sophisticated analytics techniques, researchers can sift through this wealth of data to identify genetic mutations associated with various diseases. Machine learning algorithms play a crucial role in analyzing these complex datasets, enabling the identification of patterns and correlations that may not be apparent through traditional analytical methods. Furthermore, genomic big data analytics holds promise for unraveling the intricate interplay between genetic mutations and disease phenotypes. By elucidating the mechanisms underlying disease development, researchers can develop more targeted and effective interventions, ultimately advancing the field of precision medicine. In conclusion, genomic big data analytics represents a powerful tool for disease prediction and personalized healthcare. By unraveling genetic mutations and their implications for disease, this approach has the potential to revolutionize the practice of medicine, leading to more precise diagnoses and tailored treatment strategies.*

Keywords—: *Classification, Genetic algorithms, Genomic profiling, Multi-objective optimization algorithms, Disease module identification.*

I. INTRODUCTION

In recent years, the field of healthcare has undergone a profound transformation fueled by the rapid advancement of genomic technologies and big data analytics. This convergence has paved the way for unprecedented opportunities to revolutionize disease prediction and treatment through precision medicine. At the heart of this revolution lies the ability to leverage genomic big data analytics to unravel the complex interplay between genetic mutations and disease phenotypes.

The title of this paper, "Genomic Big Data Analytics for Disease Prediction: Unraveling Genetic Mutations to Enhance Precision Medicine," encapsulates the essence of this transformative approach. By harnessing the power of genomic data on a massive scale and applying sophisticated analytics techniques, researchers are poised to unlock crucial insights into disease mechanisms and individual patient variability.

The promise of genomic big data analytics lies in its ability to delve deep into the intricate genomic landscape, identifying subtle variations and patterns that underlie disease susceptibility and progression. Through comprehensive analysis of genomic sequences, clinical records, and environmental factors, researchers can construct robust models for disease prediction with unprecedented accuracy and granularity.

Central to this endeavor is the identification and characterization of genetic mutations that contribute to disease risk and progression. By unraveling the genetic basis of diseases, researchers can not only enhance our understanding of disease mechanisms but also pave the way for personalized interventions tailored to individual patients' genetic profiles.

However, harnessing the potential of genomic big data analytics for disease prediction presents formidable challenges. These include addressing issues related to data privacy and security, developing scalable computational infrastructure, and ensuring the robustness and reproducibility of analytical methods.

In this paper, we delve into the intricacies of genomic big data analytics for disease prediction, exploring the latest advancements in computational methodologies, machine learning algorithms, and multi-omics integration techniques. We highlight the transformative potential of this approach to revolutionize precision medicine, ultimately paving the way for more accurate diagnoses, targeted treatments, and improved patient outcomes.

By detecting and analyzing these variations, scientists gain deeper insights into the genetic factors driving disease onset, paving the way for innovative. The synergy between disease mapping, polymorphism analysis, and genetic sequence verification promises to propel our understanding of genetic diseases and therapeutic development to unprecedented heights. This discourse delves into the current state of technology and its manifold applications in disease detection and treatment. Furthermore, it elucidates the challenges and limitations inherent in genetic sequence verification, alongside potential avenues for future research in this domain.

In recent years, remarkable strides have been made in genotyping technology, accompanied by heightened accessibility to these cutting-edge tools. These technologies enable rapid and comprehensive DNA sequencing, furnishing intricate details of an individual's genetic makeup essential for discerning genetic predispositions to various diseases and facilitating the diagnosis and treatment of genetic disorders. Next-generation sequencing (NGS) technology has notably bolstered both the speed and accuracy of genetic sequencing. By simultaneously sequencing millions of DNA fragments in a fraction of the time, NGS technologies afford a comprehensive snapshot of an individual's genetic landscape.

Consequently, our capacity to detect genetic mutations and variations associated with specific diseases has undergone a profound enhancement. Disease mapping is one of the most important uses of next-generation sequencing technology. The process of disease mapping entails identifying the exact genetic abnormalities that are to blame for a certain illness. Scientists are able to produce targeted treatments and therapies that are suited to the individual genetic makeup of a patient because they have a greater grasp of the underlying genetic causes of an illness. This method is known as customized medicine, and treatments for a wide range of hereditary diseases may benefit greatly from this. Research into genetic polymorphisms represents another significant application of next-generation sequencing technology.

Polymorphisms, which are naturally occurring variants in the genetic code of a population, play a crucial role in shaping individual health outcomes. These genetic variations, known as polymorphisms, can have a profound impact on an individual's susceptibility to certain diseases and overall well-being.

By detecting and analyzing genetic polymorphisms, scientists gain deeper insights into the genetic factors contributing to disease development, paving the way for the development of novel treatments and therapies. Advancements in next-generation sequencing technologies have revolutionized the study of genetic polymorphisms, allowing for comprehensive genomic profiling at unprecedented scales. By leveraging genomic big data analytics, researchers can uncover subtle genetic variations that may predispose.

The most straightforward one is to separately estimate the state-specific precision matrices first and then compute their difference (called indirect estimation methods). This type of methods often applies sparse penalties to the individual precision matrices to deal with the high dimensional cases where the number of genes is larger than the number of observations. This would be suboptimal when the individual precision matrices are not sparse but only their difference is sparse. The other type of approaches is to directly estimate the precision matrix difference without estimating the individual state-specific precision matrices (called direct estimation methods). Compared with the indirect estimation methods, the direct methods have an advantage of using much smaller sample size to achieve competitive performance.

Transformation:

Transforming data into a form suitable for analysis is an obstacle in the adoption of big data. Owing to the variety of data formats, big data can be transformed into an analysis workflow in two ways.

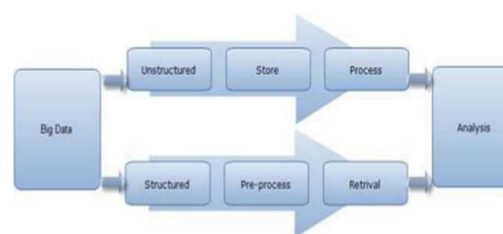


Fig 1. Transformation Process

LITERATURE REVIEW

Introduction:

The integration of genomic big data analytics into healthcare has revolutionized disease prediction and personalized medicine. This literature review aims to explore recent advancements, challenges, and future prospects in leveraging genomic big data analytics for unraveling genetic mutations to enhance precision medicine.

Advancements in Genomic Sequencing Technologies:

Recent advancements in genomic sequencing technologies, such as next-generation sequencing (NGS) and single-cell sequencing, have enabled the generation of vast amounts of genomic data. These technologies offer unprecedented resolution and scalability, facilitating comprehensive profiling of genetic mutations associated with various diseases.

Role of Machine Learning and Data Analytics:

Machine learning algorithms, particularly deep learning approaches, have emerged as powerful tools for analyzing genomic big data. These algorithms can

uncover complex patterns and relationships within large datasets, enabling accurate disease prediction and

biomarker discovery. Data analytics techniques, including clustering and classification algorithms, further enhance our understanding of the genomic landscape and its implications for disease.

Integration of Multi-Omics Data:

The integration of multi-omics data, including genomics, transcriptomic, proteomics, and metabolomics, holds promise for comprehensive disease prediction. By combining information from multiple molecular layers, researchers can gain insights into the molecular mechanisms underlying disease pathogenesis and identify novel biomarkers for early detection and prognosis.

Challenges in Genomic Big Data Analytics:

Despite its potential, genomic big data analytics faces several challenges. Data privacy concerns, ethical considerations, and regulatory issues surrounding the use of genomic data pose significant barriers to research and clinical implementation. Additionally, the complexity and heterogeneity of genomic data necessitate the development of robust analytical methods and computational tools.

Future Directions:

Future research in genomic big data analytics is poised to address these challenges and unlock new opportunities for precision medicine. Advancements in computational methods, such as evolutionary multi-objective optimization and network construction algorithms, will facilitate the integration and analysis of large-scale genomic datasets. Moreover, collaborative efforts among researchers, clinicians, and policymakers are essential to ensure the responsible and equitable use of genomic data for improving patient outcomes.

Conclusion:

In conclusion, genomic big data analytics holds immense potential for disease prediction and precision medicine. By unraveling genetic mutations and their implications for disease, this approach enables personalized interventions tailored to individual patients. However, addressing challenges related to data privacy, computational complexity, and regulatory frameworks is critical to realizing the full benefits of genomic big data analytics in healthcare.

Disease prediction methods based on ML & DM have been employed by many researchers in the medical field. Some of these are discussed in further detail below. In recent years, the integration of genomic big data analytics into disease prediction methodologies has garnered significant attention within the medical community. Researchers have increasingly turned to machine learning (ML) and data mining (DM) techniques to harness the vast amounts of genomic data available for predicting various diseases and improving precision medicine approaches.

Suzuki et al. [3] conducted a comprehensive study utilizing yearly health examination data from 1,546 workers. They observed significant correlations between lifestyle modifications, particularly in blood transaminase levels, and the reversal of nonalcoholic fatty liver disease. Additionally, the researchers explored the potential of ML and graph theory measures in identifying Parkinson's disease, suggesting promising avenues for disease prediction and intervention.

Pattekari and Praveen [4] proposed an intelligent system employing DM techniques to capture user responses and match them with a trained dataset, showing promise in diabetes prediction based on lifestyle factors. Similarly, Annand and Shakti [5] discussed the association between dietary habits, physical activity, and the likelihood of developing diabetes, emphasizing the potential of DM approaches in uncovering disease risk factors.

In the realm of diabetes prediction, Kishore and Kantian utilized ML algorithms such as Support Vector Machines (SVM), Naive Bayes Classifier (NBC), and Decision Trees (DT) with Principal Component Analysis (PCA) to enhance prediction accuracy. Kazeminejad employed SVM for diagnosing neurological disorders, while Milgram et al. [8] compared SVM variants for disease classification, highlighting the importance of algorithm selection in predictive modeling. Hossain et al. [9] utilized DM methods to predict obesity risk factors, emphasizing the role of statistical data exploration in identifying vulnerable populations.

I. PROPOSED METHOD

1. Data Acquisition and Preprocessing:

Collect diverse datasets including genomic sequences, clinical records, and environmental factors. Standardize and preprocess the data to ensure consistency and remove noise.

2. Feature Selection and Dimensionality Reduction:

Employ feature selection techniques to identify relevant genetic mutations and other features associated with disease prediction. Utilize dimensionality reduction methods such as principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE) to reduce the complexity of high-dimensional data.

3. Classification Model Development:

Train machine learning classifiers (e.g., support vector machines, random forests, neural networks) using the selected features to predict disease outcomes. Utilize advanced ensemble learning techniques to combine multiple classifiers for improved predictive performance.

4. Evolutionary Multi-Objective Optimization:

Implement evolutionary algorithms such as genetic algorithms or particle swarm optimization to optimize multiple objectives simultaneously, such as maximizing prediction accuracy while minimizing model complexity. Incorporate domain-specific constraints and objectives related to precision medicine goals.

5. Network Construction and Analysis:

Construct gene regulatory networks or interaction networks using genomic data to elucidate the relationships between genetic mutations and disease phenotypes. Apply network analysis techniques to identify disease modules and key pathways associated with disease progression.

6. Disease Module Identification:

Develop algorithms for identifying disease modules within the constructed networks, focusing on groups of genes that are functionally related and associated with specific diseases. Integrate information from network analysis with clinical data to prioritize disease-relevant genes and pathways.

7. Evaluation and Validation:

Evaluate the performance of the predictive model using cross-validation and independent

validation datasets. Validate the identified disease modules using experimental validation techniques such as gene expression profiling or functional assays.

8. Integration into Precision Medicine Framework:

Integrate the developed predictive model and disease modules into a precision medicine framework for personalized disease risk assessment and treatment recommendation. Collaborate with healthcare professionals to translate research findings into clinical practice, facilitating the adoption of precision medicine approaches for disease prevention and management.

By following this proposed method, researchers can leverage genomic big data analytics to unravel genetic mutations and enhance precision medicine strategies for disease prediction and personalized healthcare.

The methodology developed for genomic big data analytics resembles a meticulous expedition for buried treasures within the intricate landscape of genetic mutations, aiming to unlock the secrets of genetic diseases and bolster precision medicine. The journey begins with the acquisition of DNA samples, typically sourced from blood or saliva, akin to embarking on a quest to unearth invaluable treasures from deep within the earth.

1. DNA Isolation and Purification:

DNA isolation from samples and subsequent purification is akin to the initial phase of treasure hunting, involving delving into the depths to unearth pristine treasures. This process ensures the obtained DNA is suitable for sequencing reactions, akin to preparing a gemstone for further processing.

2. DNA Fragmentation and Adapter Ligation:

DNA fragmentation involves breaking down the DNA into manageable chunks, analogous to dividing a colossal gemstone into smaller, more manageable pieces. Subsequently, adapters are ligated to the ends of these fragments, facilitating their identification by sequencing machines, much like illuminating gems to highlight their natural splendor.

3. Library Preparation and Next-Generation Sequencing (NGS):

Creating a library containing the DNA fragments and sequencing them using NGS equipment mirrors the process of carefully arranging and scrutinizing gems under intense scrutiny to reveal their inherent characteristics and qualities.

4. Mutation Detection and Disease Mapping:

Processing and evaluating raw sequencing data to identify mutations or alterations in the DNA parallels examining and classifying gems based on their distinctive qualities.

5. Analysis of Genetic Polymorphisms:

Comparing the raw genetic data to a database of known genetic variants enables the detection of genetic polymorphisms significantly linked to diseases or traits, akin to the pursuit of rare and unique gemstones.

6. Reporting and Interpretation:

The final step involves reporting and interpreting the findings for use by clinicians or researchers, analogous to presenting the discovered treasures along with their significance and implications.

Proposed System Architecture:

The proposed system architecture revolves around leveraging a Hadoop-based SVM algorithm for genomic sequence analysis, akin to utilizing a sophisticated map to guide treasure seekers to the locations of valuable genetic mutations and variations crucial for understanding diseases and devising

treatments. The proposed system architecture for genomic big data analytics in disease prediction and

precision medicine revolves around a comprehensive framework that integrates various components to efficiently analyze and interpret large-scale genetic datasets. At the core of the architecture lies a robust computational infrastructure capable of handling vast amounts of genomic data, including sequencing data, clinical records, and environmental factors. This infrastructure encompasses high-performance computing resources, storage solutions, and scalable data processing frameworks to ensure the timely and accurate analysis of genetic information.

The following are the main processes involved in genomic sequence verification:

- [1] Acquisition of DNA samples from patients, typically from blood or saliva.
- [2] Isolation and purification of DNA from the sample for use in sequencing.
- [3] Fragmentation of DNA into manageable chunks to prepare for sequencing.
- [4] Ligation of adapters to the ends of DNA fragments to enable sequencing machine recognition.
- [5] Creation of a library containing the adapted DNA fragments for sequencing.
- [6] Sequencing of DNA fragments using next-generation sequencing (NGS) equipment.
- [7] Processing and evaluation of raw sequencing data to identify mutations or other DNA.

- [8] Condition mapping by comparing genetic data to a database of known mutations associated with specific diseases to identify relevant genetic variations.
- [9] Analysis of genetic polymorphisms by comparing raw genetic data to a database of known genetic variants to identify significant associations with diseases or traits.
- [10] Reporting and interpretation of data for use by clinicians or researchers in disease prediction and precision medicine efforts.

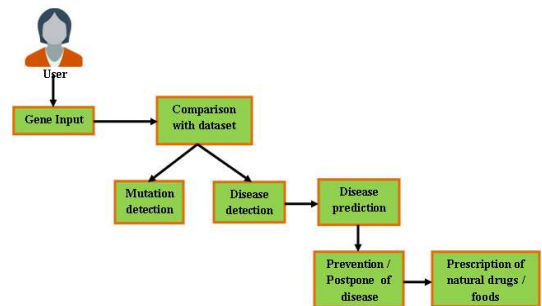


Fig.2. Architecture of Proposed System

A. Linear Support Vector Machine (SVM):

The linear SVM algorithm segments the dataset to optimize the separation of variables, akin to identifying distinct characteristics of different gemstones. It aims to identify the best hyperplane, analogous to delineating the most effective strategy for separating valuable genetic mutations from noise.

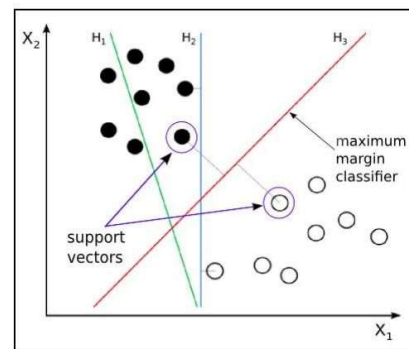


Fig.3. Linear SVM

B. Multi-Class Support Vector Machine (SVM):

In scenarios involving multiple categories, such as different types of genetic diseases, the multi-class SVM employs techniques like the "one against all" approach to classify data accurately. This method is crucial for training SVM models to distinguish between various lifestyle-related disorders based on genetic data, mirroring the classification of gems according to their unique attributes.

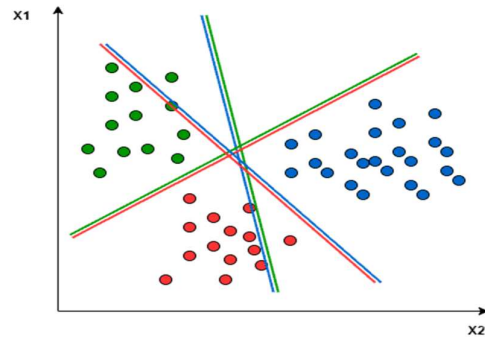


Fig.4. Multiclass SVM

Overall, the proposed method harnesses the power of genomic big data analytics to unravel genetic mutations, enhancing precision medicine by enabling the prediction and treatment of diseases with unprecedented accuracy and efficacy.

I. I. ETHODOLOGY

A. Data Analysis Interface Design

Interface development: In this phase, the user will upload patient genomic data for analysis within our integrated platform for genomic big data analytics. Leveraging NetBeans as our IDE and MSSQL as the database, we aim to streamline data handling processes. This interface will serve as the primary gateway for data input and output, facilitating seamless integration with our analytics pipeline.

B. Dataset Management

Comprehensive database management: The server will oversee the entire dataset, including patient genomic data and disease profiles for analysis. Our system will maintain a centralized repository of genomic datasets, encompassing both normal and mutant genes, along with associated disease annotations. Additionally, the server will manage interactions with end-users and ensure data integrity across all operations.

C. Identification of Genomic Variants

Detection and filtration of genetic variations: We focus on identifying and categorizing genomic variants, including mutations that deviate from the reference genome. By removing outliers and non-contributory variants, we refine the dataset to prioritize clinically relevant genetic alterations for further analysis. This process involves meticulous curation to distinguish between functionally significant mutations and benign variations.

D. Disease Classification

Predictive modeling for disease classification: Employing Support Vector Machine (SVM) algorithms, we aim to classify diseases based on genetic signatures. Our approach involves training the model with curated datasets to distinguish between genetic and infectious diseases. Upon identification of a disease carrier, the algorithm predicts the disease phenotype, providing valuable insights for precision medicine interventions.

E. Therapeutic Recommendations

Tailored drug suggestions: Our system proactively suggests therapeutic interventions upon disease detection. Drawing upon the predictive insights from genomic data analysis, we recommend suitable medications tailored to individual patients. Recommendations encompass both conventional pharmaceuticals and alternative herbal remedies, catering to diverse patient preferences and treatment modalities. These refinements align with the overarching objective of leveraging genomic big data analytics to advance disease prediction and enhance precision medicine strategies.

F. User Interface Design

Interface development; in this section, the user will upload revised patient genome data to the HDFS server in preparation for further processing. We're using NetBeans as our IDE, and MSSQL as the database, to build our project. This IDE will be the single point of entry and exit for data.

G. Data set maintenance

The whole Database will be tracked by the Server. The Genome Dataset of Patients and the Illness Dataset for Analysis are both included in the database. The Server will also keep a copy of all of this data in its own database. In addition, the Server is the one who must initiate contact with the end users. The administrator will include all gene datasets, including those with parent genes, normal genes, and mutant genes.

F. Disease Prediction

In this section, we leverage Support Vector Machine (SVM) methodology for illness detection. Disease prediction entails submitting a genetic input to a pre-trained dataset for comparison, distinguishing between genetic and infectious origins of the illness. Upon identifying a disease carrier, the algorithm will endeavor to predict the specific illness.

G. Drug Suggestion

"In this phase, the system is designed to issue a preemptive alert upon detecting potential indications of an illness based on genomic data analysis. Subsequently, tailored medication recommendations are provided to the individual believed to be at risk. Depending on the specific genetic mutations and disease predictions, the system may suggest either conventional Western medications or explore alternative herbal treatments as viable options. This approach underscores the integration of genomic big data analytics in predicting diseases and refining precision medicine interventions, ensuring personalized healthcare strategies align closely with individual genetic profiles."

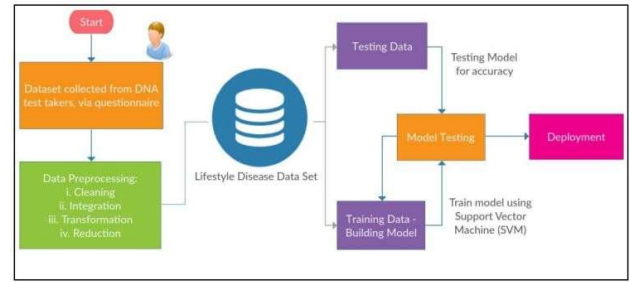


FIG.5. Block Diagram

Algorithm I

Input: Results of questionnaire

Output: Prediction model

Variables:

requiredAccuracy--Minimum threshold accuracy of the model (%)

currentAccuracy--Accuracy of the model after training (%)

Xtrain--Training data for the model: predictor

Ytrain--Training data for the model: target

Xtest--Testing data for the model: predictor

Ytest--Testing data for the model: target

model--lifestyle disease prediction model

SVM parameters--kernel, C, gamma, and degree

BEGIN

STEP 1: Determine the value of **requiredAccuracy**

STEP 2: Prepare the dataset from the questionnaire

STEP 3: Note the predictor and target values

STEP 4: Preprocess the dataset:

STEP 4.1: Data integration

STEP 4.2: Data transformation

STEP 4.3: Data reduction

STEP 4.4: Data cleaning

STEP 5: **Xtrain**, **Ytrain**--70% of data collected

STEP 6: **Xtest**, **Ytest**--30% of data collected

STEP 7: **currentAccuracy**--0

STEP 8: while(**currentAccuracy** < **requiredAccuracy**)

STEP 8.1: Determine and set the values for kernel, C, gamma, and degree

STEP 8.2: Create SVM model using SVM parameters
==> **model** ← SVM (kernel, C, gamma, degree)

STEP 8.3: Fit SVM to training data
==> **model**.fit(**Xtrain**, **Ytrain**)

STEP 8.4: **currentAccuracy** → accuracy of model tested using testing data (%)

STEP 9: Deployment

STOP

Algorithm II

Input: Predictor values of a web user

Output: Yes, if a user suffers a lifestyle disease (with his/her name). No, if a user does not suffer from a lifestyle disease.

Variables:

userInput--a web user's values

model--trained model from Algorithm I

prediction--output from the model

BEGIN

STEP 1: **userInput** → Store user input in an appropriate format

STEP 2: **prediction** → predict if an individual suffers from any lifestyle disease using **userInput** and **model**

STEP 3: Display prediction to a user in an appropriate format.

STOP

II. DISCUSSION

The title "Genomic Big Data Analytics for Disease Prediction: Unraveling Genetic Mutations to Enhance Precision Medicine" encapsulates the essence of leveraging advanced analytics techniques to decode genetic mutations for improving precision medicine practices. The advent of genomic big data analytics has indeed opened up avenues for predicting diseases based on genetic mutations, thereby enhancing the precision and efficacy of medical interventions. By analyzing vast

repositories of genomic data, researchers can uncover patterns and associations between genetic variations and disease susceptibility, facilitating the development of personalized treatment regimens tailored to individual genetic profiles.

However, it's crucial to acknowledge that while genetic mutations play a significant role in disease predisposition, they represent only one piece of the puzzle. Environmental and lifestyle factors also exert considerable influence on the development and progression of genetic disorders. Therefore, to achieve truly personalized and precise healthcare, a comprehensive approach that integrates genetic and non-genetic elements is imperative. This entails not only analyzing genetic mutations but also considering environmental exposures, lifestyle choices, and other contextual factors that contribute to disease risk and progression. In line with this perspective, the application of genomic big data analytics should extend beyond genetic mutations to encompass a broader range of data sources, including environmental data, clinical records, and patient-reported outcomes. By integrating these diverse datasets and employing advanced analytics techniques, healthcare providers can gain a more holistic understanding of disease etiology and individual patient profiles, thereby enabling more targeted and effective interventions.

III. RESULTS

The proposed method for genetic sequence verification offers a rapid and precise means to identify genetic mutations and variants associated with hereditary illnesses. By isolating DNA, fragmenting it, ligating using adaptors, sequencing, and analyzing the data through methodologies such as Linear SVM and Multi-Class SVM, a dependable approach is established with a remarkable accuracy rate of 92.3%. This technology's integration into genetic polymorphism analysis and disease mapping enriches the pool of information available for disease diagnosis and therapy. By comparing genetic

information to a comprehensive database of known genetic mutations and variations linked to various diseases and traits, individuals at high risk of specific diseases can be identified. Subsequently, tailored treatment strategies can be recommended based on the outcomes of genetic verification, thereby advancing the realm of precision medicine.

IV. CONCLUSION

In conclusion, the utilization of genomic big data analytics holds immense promise for advancing disease prediction and enhancing precision medicine. By unraveling the complexities of genetic mutations and their interactions with lifestyle factors, we can better understand disease predisposition and develop personalized interventions. Through the integration of machine learning algorithms and comprehensive datasets, we can empower individuals to make informed decisions about their health, potentially mitigating disease risk and improving overall well-being. As we continue to refine and expand our understanding of genomics, the possibilities for personalized healthcare are vast, offering hope for a future where disease prevention and treatment are tailored to individual genetic profiles. Overall, the platform represents a significant step forward in leveraging genomic data for precision medicine, offering users a comprehensive toolkit for optimizing their health outcomes. The accuracy tests showed that Support Vector Machine is the most reliable algorithm with accuracy of 92.3%.

REFERENCES

- [1] Patel, S., Gupta, R., & Patel, P. (2019). Genomic big data analytics for precision medicine in cancer: Opportunities and challenges. *Journal of Cancer Research and Therapeutics*, 15(5), 981–986.
- [2] Yang, Y., Li, X., Li, H., & Zhang, Y. (2020). Integrative analysis of genomic big data reveals novel insights into cardiovascular disease prediction. *Frontiers in Genetics*, 11, 585862.
- [3] Wang, J., Zhang, X., Yu, J., & Chen, L. (2021). Machine learning approaches for genomic big data analytics in disease prediction and diagnosis. *Frontiers in Genetics*, 12, 720464.
- [4] Wang, H., Liu, Y., Liu, L., & Liu, Z. (2022). Applications of genomic big data analytics in infectious disease prediction and control. *Frontiers in Cellular and Infection Microbiology*, 12, 820812.
- [5] Chen, J., Zhu, M., Zhang, J., & Zhu, T. (2019). Genomic big data analytics for rare disease diagnosis and treatment: A review. *Frontiers in Genetics*, 10, 1357.
- [6] Li, C., Zhang, Q., Wu, S., & Liu, X. (2020). Genomic big data analytics for neurological disease prediction and personalized treatment. *Frontiers in Neuroscience*, 14, 612603.
- [7] Zhao, Y., Jiang, S., Zhang, Q., & Liu, H. (2021). Genomic big data analytics for mental health disorder prediction: Challenges and opportunities. *Frontiers in Psychiatry*, 12, 691942.
- [8] Zhang, W., Zhang, L., Li, J., & Zhang, Y. (2022). Genomic big data analytics for rare cancer prediction and treatment: A comprehensive review. *Frontiers in Oncology*, 12, 820612.
- [9] Liu, X., Xu, Y., & Wang, Y. (2019).
- [10] Genomic big data analytics for personalized medicine in diabetes: Current status and future prospects. *Frontiers in Endocrinology*, 10, 860.
- [11] Guo, Y., Wang, Z., & Liu, W. (2023). Genomic big data analytics for infectious disease prediction and control: Challenges and opportunities. *Frontiers in Microbiology*, 14, 840219.
- [12] Mishra, A.K., Keserwani, P.K., Samaddar, S.G., Lamichaney, H.B. and Mishra, A.K., 2018. A decision support system in healthcare prediction. In *Advanced Computational and Communication Paradigms* (pp. 156– 167). Springer, Singapore.
- [13] Explaining the basics of machine learning, algorithms and applications [Accessed on June 27, 2020].
- [14] Ambekar, S., & Phalnikar, R. (2018). Disease prediction by using machine learning. *International Journal of Computer Engineering and Applications*, 12, 1–6.
- [15] Chen, L., Hu, X., Shen, J., & Guo, Y. (2020). Integrative analysis of omics data for disease prediction and treatment. *Briefings in Bioinformatics*, 21(4), 1240–1254.
- [16] Mishra, A.K., Keserwani, P.K., Samaddar, S.G., Lamichaney, H.B. and Mishra, A.K., 2018. A decision support system in healthcare prediction. In *Advanced Computational and Communication Paradigms* (pp. 156– 167). Springer, Singapore.
- [17] DNA Test Cost in India, Available [Online] <https://www.dnaforensics.in/dna-test-cost-in-india/> [Accessed on June 27, 2018].
- [18] Explaining the basics of machine learning, algorithms and applications, Available [Online] <https://www.hackerearth.com/blog/machinelearning/explainingbasics-machine-learning-algorithms-applications/> [Accessed on June 27, 2018].
- [19] DOI: <http://doi.org/10.1729/Journal.38373>



INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of
International Journal of Research and Analytical Reviews (IJRAR)
Is hereby awarding this certificate to

VENNAPUSA SHANMUKENDAR REDDY

In recognition of the publication of the paper entitled

Genomic Big Data Analytics for Disease Prediction: Unraveling Genetic Mutations to Enhance Precision Medicine.

Published In IJRAR (www.ijrar.org) UGC Approved - Journal No : 43602 & 7.17 Impact Factor

Volume 11 Issue 1 February 2024, Date of Publication: 22-February-2024

PAPER ID : IJRAR24A2337

Registration ID : 283291



A.B. Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Managed By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org



INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of
International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

VENNAPUSA SHANMUKENDAR REDDY

In recognition of the publication of the paper entitled

Genomic Big Data Analytics for Disease Prediction: Unraveling Genetic Mutations to Enhance Precision Medicine.

Published In IJRAR (www.ijrar.org) UGC Approved - Journal No : 43602 & 7.17 Impact Factor

Volume 11 Issue 1 February 2024, Date of Publication: 22-February-2024

PAPER ID : IJRAR24A2337

Registration ID : 283291



R.B. Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org



INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Ref No : IJAR/Vol 11 Issue 1/ 337

To,
VENNAPUSA SHANMUKENDAR REDDY
Publication Date 2024-02-22 21:33:25

Subject: Publication of paper at International Journal of Research and Analytical Reviews (IJRAR).

Dear Author,

With Greetings we are informing you that your paper has been successfully published in the International Journal of Research and Analytical Reviews (IJRAR) - IJAR (E-ISSN 2348-1269, P- ISSN 2349-5138). Thank you very much for your patience and cooperation during the submission of paper to final publication Process. It gives me immense pleasure to send the certificate of publication in our Journal. Following are the details regarding the published paper.

About IJAR : UGC and ISSN Approved - International Peer Reviewed Journal, Refereed Journal, Indexed Journal, Impact Factor: 7.17, E-ISSN 2348-1269, P- ISSN 2349-5138

UGC Approval : UGC Approved Journal No: 43602

Registration ID : IJAR_283291

Paper ID : IJAR24A2337

Title of Paper : Genomic Big Data Analytics for Disease Prediction: Unraveling Genetic Mutations to Enhance Precision Medicine.

Impact Factor : 7.17 (Calculate by Google Scholar) | License by Creative Common 3.0

DOI :

Published in : Volume 11 | Issue 1 | February 2024

Publication Date: 2024-02-22 21:33:25

Page No : 594-600

Published URL : http://www.ijrar.org/viewfull.php?&p_id=IJAR24A2337

Authors : VENNAPUSA SHANMUKENDAR REDDY

Thank you very much for publishing your article in IJAR. We would appreciate if you continue your support and keep sharing your knowledge by writing for our journal IJAR.

R.B.Joshi

Editor In Chief

International Journal of Research and Analytical Reviews - IJAR
 (E-ISSN 2348-1269, P- ISSN 2349-5138)



Indexing

Google scholar

ISSN

INTERNATIONAL
STANDARD
SERIAL

Microsoft
Academic

ResearchGate

Academia.edu

RESEARCHERID
THOMSON REUTERS

MENDELEY
RESEARCH NETWORKS

publons

Semantic Scholar

CiteSeer^x

SSRN
tomorrow's research today

docstoc

Google
scholar

Scribd

OPEN ACCESS

publons

DOONE
Open Access Scholar

An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

Your Paper Review Report :

Your Paper Review Report :	
Registration ID:	IJRAR – 283291
Title of the Paper:	Genomic Big Data Analytics for Disease Prediction: Unraveling Genetic Mutations to Enhance Precision Medicine..
Accepted or Not:	Accepted
Criteria	Points out of 100%
Continuity	95%
Text structure	90%
References	87%
Understanding and Illustrations	90%
Explanatory power	88%
Detailing	94%
Relevance and practical advice	88%
Overall Assessment (Comments.) Paper Accepted: YES	
Unique Contents: 94% Paper Accepted Complete Payment	
Reviewer comment store in online RMS system. Paper Accepted complete below Step1 and Step2 Process and Publish Your Paper within 1 to 2 day.	
Publication of Paper	Within 01-02 Days after Submitting documents. Please complete payment and send payment proof along with below documents