National College of Ireland

Effective use of Cloud Computing and Machine Learning Technologies for Smart Healthcare Applications

MSc Research Project
MSc in Cloud Computing

Suraj Beragu
Student ID: x21117951

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Suraj Beragu
Student ID:	X21117951
Programme:	MSc in Cloud Computing **Year:** 2022-23
Module:	Research Project
Supervisor:	Sean Heeney
Submission Due Date:	15th December 2022
Project Title:	Effective use of Cloud Computing and Machine Learning Technologies for Smart Healthcare Applications
Word Count:	9201 **Page Count:** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: 

Date: 15th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Effective use of Cloud Computing and Machine Learning Technologies for Smart Healthcare Applications

Suraj Beragu
x21117951
MSc in Cloud Computing
National College of Ireland

ABSTRACT

Background: The Pandemic has triggered the Healthcare sector to look at effective use of technology to enhance the delivery of patient care, this includes use of latest technologies for delivering healthcare applications. The use of legacy and traditional technologies for deploying health care applications causes significant challenges in today's day and age. The use of technologies like cloud computing and machine learning can provide the healthcare sector a highly scalable and reliable solution.
Objectives: The Main objective of this research is to Develop a Web application using Python (Django) framework with machine learning model for disease prediction. Which could lead the healthcare sector into a more functional and scalable application architecture.
Methodology: The development of the web application is performed using Python-Django Framework and the necessary dataset is obtained from Kaggle. The use of AWS services for Continuous Integration and Continuous deployment (CI/CD) with Code Pipeline, Code Build and Elastic Beanstalk with continuous integration with Git. For the machine learning model Random Forest, K-Nearest Neighbor and Convolutional Neural network classifiers are used. The use of fog and edge computing paradigms are also observed as part of this research.
Results: The deployment of the Django based Web Application with machine learning model to predict diseases with full CI/CD lifecycle is shown and the the machine learning model provides the prediction with high accuracy.
Findings: The methodology used was seen to be successful in terms of deploying the application with ML Model on AWS cloud and the machine learning model could predict diseases with the symptoms of the user/patient.

Keywords: Cloud Computing, Fog and edge computing, Machine Learning, HealthCare

1. INTRODUCTION

The emergence of cloud computing and machine learning has greatly improved the efficiency of healthcare systems. By leveraging the power of cloud computing and machine learning, healthcare systems can develop more efficient methods of providing better care to their patients while also reducing the cost of healthcare delivery. This report explores how cloud computing and machine learning can be used to develop an efficient healthcare system. Specifically, this report looks at how cloud computing and machine learning can be used to improve the accuracy and speed of disease prediction, as well as how cloud computing and machine learning can be used to improve the efficiency of healthcare application delivery. In the current healthcare system, cloud and machine learning can be used to develop an efficient healthcare system. The use of cloud computing and machine learning can help in the areas of disease prediction, task offloading, and CI/CD with Jenkins Python (Django). This paper will discuss the potential of cloud and machine learning for the development of an efficient healthcare system.

The Potential of Cloud Computing

Cloud computing can be used to develop an efficient healthcare system by providing a platform for storing and processing large amounts of data. Data can be stored in a cloud-based system, which can then be used to analyze data and make decisions. Additionally, cloud computing can allow for the sharing of data between different healthcare providers, allowing for better collaboration and coordination of care.

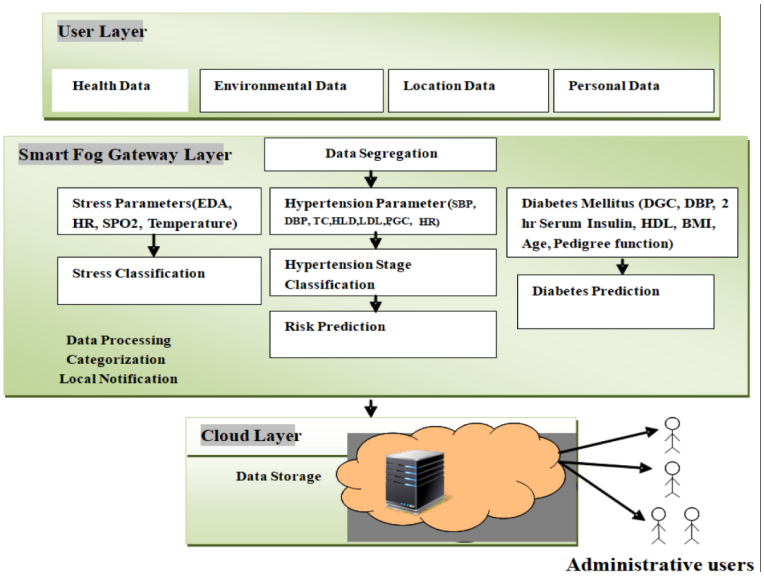

Figure 1: The figure shows a high-level architecture diagram of different computing layer

The Potential of Machine Learning for healthcare sector

Machine learning can be used to develop an efficient healthcare system by providing a platform for predictive analysis. Machine learning algorithms can be used to analyze patient data to identify patterns and trends in the data. This can be used to make predictions about patient outcomes and to identify potential treatments that may be effective. Additionally, it also can be used to identify high-risk patient populations and to provide personalized treatments that are tailored to each patient's needs.

Cloud Computing and Disease Prediction

Cloud computing provides a platform for sharing and storing data. By leveraging the power of cloud computing, healthcare providers can access and store large amounts of data, including medical records, which can be used to improve the accuracy of disease prediction. By using machine learning algorithms, such as neural networks and deep learning, healthcare providers can develop models to accurately predict diseases. These models can be trained and tested on large datasets stored in the cloud, allowing healthcare providers to predict diseases quickly and accurately.

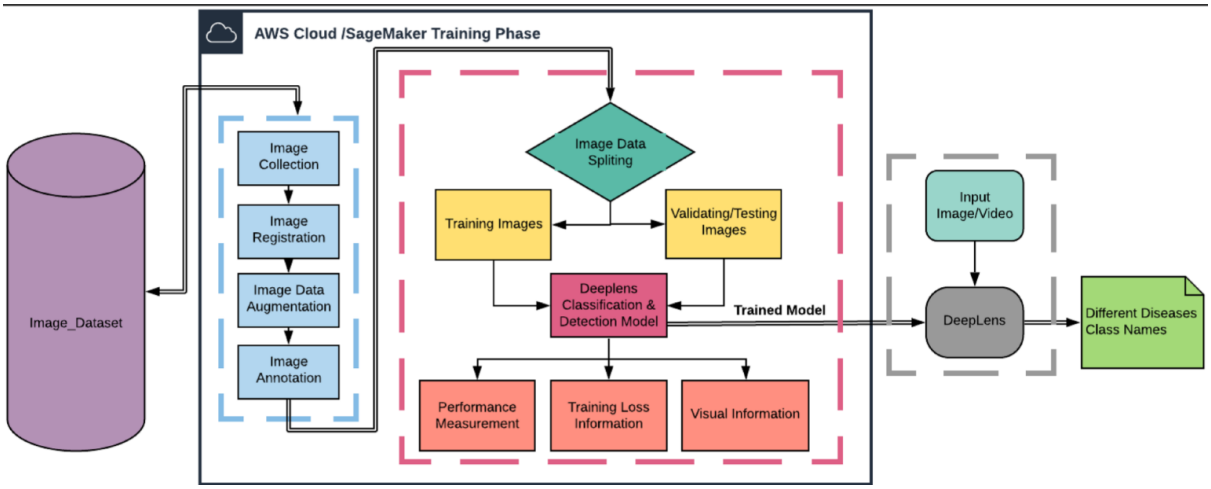

Figure 2: The figure shows an architecture diagram of Machine learning model deployment in AWS Sage maker

Edge Computing and Disease Prediction

Edge computing is a type of computing that enables data processing and computation at the edge of a network, near the source of data. By leveraging the power of edge computing, healthcare providers can quickly process data and make decisions in real time. This can be used to improve the accuracy of disease prediction. Edge computing can be used to quickly process and analyze data from medical devices, such as medical imaging devices and medical sensors, allowing healthcare providers to predict diseases quickly and accurately.

Cloud Computing and Healthcare Delivery

Cloud computing can also be used to improve the efficiency of healthcare delivery. By leveraging the power of cloud computing, healthcare providers can access and store large amounts of data, such as patient records and medical images. This data can be used to develop models that automate healthcare processes, such as scheduling appointments and ordering tests. These models can be deployed in the cloud, allowing healthcare providers to deliver healthcare services quickly and efficiently to their patients.

Cloud computing and machine learning are two of the most powerful and emerging technologies which have the potential to revolutionize the healthcare industry. Cloud computing enables data storage and communication, while machine learning allows for predictive analytics and automated decision-making. By combining these two technologies, it is possible to develop an efficient healthcare system that can provide better diagnosis and treatment of diseases, improve patient outcomes and reduce costs.

Cloud computing can be used to store and share patient data, such as medical records and lab results, securely. It can also be used for communication between different hospitals, clinics, and pharmacies, allowing for faster and more efficient patient care. Furthermore, cloud computing can be used to store and analyze large amounts of medical data, allowing for personalized healthcare decisions and disease detection.

Machine learning can be used to build predictive models for diagnosing and predicting diseases. These models use data from medical records, imaging tests, and other sources to analyze patterns and predict the likelihood of a patient having a certain disease. The models can also be used to predict which treatments and therapies would be most effective for a particular patient.

These technologies can also be used to create an efficient healthcare system that can track patient health and provide personalized treatments. For example, machine learning can be used to detect early signs of disease, such as changes in vital signs, and alert physicians if necessary. By using cloud computing, healthcare providers can access patient data from anywhere, allowing them to provide better and faster care.

In addition, cloud computing and machine learning can be used to develop an efficient healthcare system that can provide preventative care. For example, machine learning can be used to detect trends in health data and alert patients when certain symptoms are present, allowing for early detection and treatment of diseases.
Finally, cloud computing and machine learning can be used to develop an efficient healthcare system that can track and monitor patient health over time. This can be used to detect changes in health, alert physicians when necessary, and provide personalized treatments to patients.
The potential of cloud computing and machine learning to revolutionize healthcare is immense. In the future, these technologies will be used to develop even more efficient healthcare systems that can provide better diagnosis and treatment of diseases, improve patient outcomes, and reduce costs.

The Use of CI/CD with Django Application with machine learning model

CI/CD with Python (Django) web application with machine learning model to predict diseases can be used to develop an efficient healthcare system by automating the development and deployment of applications. This can help to reduce the time it takes to deploy applications, as well as reduce the amount of manual effort required to develop and deploy applications. Additionally, CI/CD with Jenkins Python (Django) can help to ensure that applications are consistently and properly tested before they are deployed, helping to reduce the chances of errors occurring in the production environment.

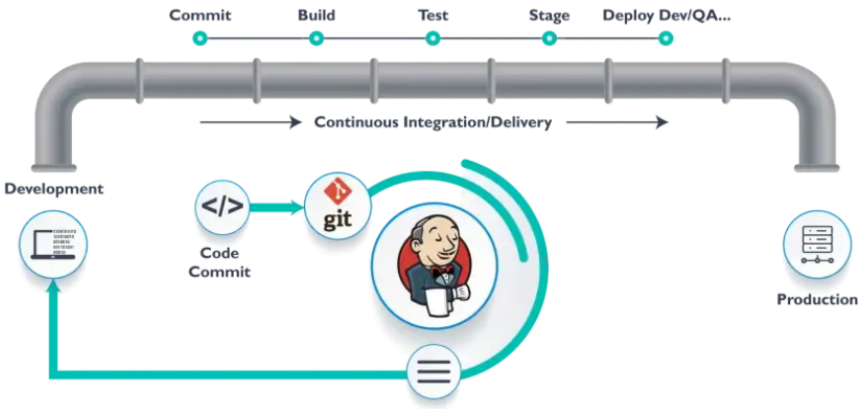

Figure 3. The diagram shows the different stages in the CI/CD Pipeline

2. PROJECT REQUIREMENT SPECIFICATION

2.1 Research Question

"How effectively can the cloud and machine learning technologies be used to enable the healthcare sector"

2.2 Motivation

The current state of the healthcare sector is decentralized where the communication between the healthcare entities like hospitals, pharmacies and the pharmaceutical companies is lacking and also the real time analysis and forecasting of diseases at an early stage is required to effectively predict and forecast diseases. Every year, several critical diseases like heart disease, liver disease and different forms of cancer claim the life of close to a million individuals, where almost half of them are abrupt, sudden and without any warning of symptoms. It is seen that in several previous studies were limited to analyzing the impact of machine learning approaches in disease prediction. However, this work is more about the use of cloud computing and machine learning technologies would play a huge role for the growth of the healthcare sector.

2.3 Discussion

This research aims to present an effective use of using cloud and edge computing for healthcare systems, with a focus on using Amazon Web Services (AWS) for Continuous Integration/Continuous Delivery (CI/CD) with Git, Python (Django) for development of the web application, and machine learning algorithms for disease prediction. This implementation would allow healthcare systems to improve their data processing capabilities and better detect, diagnose, and treat diseases

2.4 Literature review

In this section, we examine the various research studies pertaining to deployment strategies for healthcare application architecture, including the interoperability between fog and edge nodes and the cloud, as well as the security and privacy measures adopted, as well as a critique of the machine learning models used to predict diseases. The categorization of this section is as follows:

1) The use of cloud, fog and edge computing technologies for effective deployment of healthcare applications.
2) A review of the machine learning models for effective disease prediction

In the past decade, machine learning-based models have been used widely to forecast diseases and the probability of contracting a disease. In the field of medicine, these models can be utilized to detect a variety of diseases with a high degree of precision in a relatively short amount of time. This section provides an analysis and evaluation of the machine learning models that earlier researchers employed in their disease prediction efforts.

There has been multiple researchers who talked about the use of cloud and fog and edge computing is empirical to the healthcare sector. The use of these technologies lays a primary foundation to the growth of the healthcare sector in terms

of operational capacity and also to have a more advanced telemedicine framework. Here, we see the role of these frameworks, which can be classified as follows –

1) A large number of benefits can be obtained by having a fog and edge computing architecture which allows a low latency network for real time monitoring of patients.

2) Processing of large amounts of data need the interoperability with cloud and the fog nodes, where the resource intensive jobs can be pushed to the cloud and real time transactions are maintained at the fog and edge nodes.

3) Privacy and Security of data at both fog nodes and on the cloud to maintain user data privacy and confidentiality.

2.4.1 Review of the cloud, fog and edge computing technologies for effective deployment of healthcare applications.

To expand IoT systems, workloads must be moved from devices with limited capabilities to a broader pool of resources, emphasizing the need of task offloading in edge contexts. Mobile devices make ensuring service continuity difficult due to their mobility. As a consequence, outsourcing work to a more reputable organization is a viable choice. The authors identified Mobile Edge Computing (MEC) as a solution for offloading tasks in a mobile setting. Wang et al.

Zhao et al.talked about computational offloading techniques for mobile devices. They focused on tasks that need a lot of computing power and therefore use more energy. Since there are more and more apps that need a lot of processing power, mobile devices need smart ways to decide which tasks to run locally and which ones to move to the cloud. The authors say that most offloading is done to a fog node or a cloud. Both of these things have pros and cons. For instance, the cloud has a lot of resources but is usually far from mobile nodes. Fog, on the other hand, is close by but doesn't have as many resources as a cloud. So, sending work to a cloud or a fog uses different amounts of energy and gives different computation gains. In this case, the authors suggested an algorithm that minimized the amount of energy used when a task was offloaded. First, they figure out how much energy is used when the task is sent to the fog instead of the cloud. Then, they decided which entity was better based on how much computing was needed for each task. Based on these factors, the job is given to the person or group that wants it.

Mohapatra et al. came up with a semi-hybrid architecture that uses a sensor cloud. This is needed for monitoring patients from far away while keeping the network flow going. By using the sensor cloud to check on a patient's health, which can be shown in their proposed system, they were able to get some benefits. The writers did talk about cloud computing and how it could be used to help healthcare organizations collect data from patients. Sensors are used in different systems that are attached to medical equipment to collect information about patients. This information is then sent to the cloud to give restricted access.

2.4.2 Role of machine learning in edge computing and healthcare

Machine learning lets systems automatically learn programs from data, making machines smart and reducing the amount of work that needs to be done by hand. Machine learning can do predictive analysis and data mining (a more advanced version of which is called "deep learning"), which are very important for making smart healthcare apps. Machine learning-based techniques must first classify the data in order for the system to "learn" or become smart enough to make accurate predictions about changes in the system (whether it's healthcare or something else). In the machine learning system, a part called the "classifier" does the sorting. The feature values are put into the classifier, and the class is what comes out of it.

According to Hussain et al., machine learning will be one of the primary drivers of the majority of the breakthroughs that will occur in the foreseeable future. Learning by machine will play a significant part in the development of technology in a wide variety of fields, including the healthcare industry, where it will be used to perform tasks such as disease prediction, emergency detection, and drug discovery. The most important aspects of applying machine learning to the field of healthcare would be data mining and predictive analysis. According to the authors, there are a variety of approaches to machine learning; nevertheless, the classifier is the approach that has gained the most traction in recent years. When using a classifier, a vector of continuous values is fed into the system. The system then analyzes these values and classifies them by delivering a discrete value that corresponds to the type of data that was fed into the system.

The authors Chen et al. place a strong emphasis on the significance of reliable analysis of medical data and disease prediction based on machine learning for the purposes of patient care and community service. The authors noted that when there is a lack of data, it makes it difficult to accurately anticipate sickness. The authors of the study offered a latent factor model in order to reconstruct the parts of the data that were absent as a solution to this problem. The authors suggested a

disease risk prediction model based on a convolutional neural network (CNN) by utilizing structured and unstructured actual hospital data from 2013 to 2015. The time period covered by the study was 2013 to 2015.

2.4.3 Review of the Machine learning models used to predict diseases –

A review on heart disease prediction

Heart disease is frequently recognized as one of the most severe illnesses that may affect humans. This problem arises when the heart is incapable of pumping sufficient blood to the body's organs. In order to successfully prevent and treat any type of heart failure, it is vital to diagnose the condition accurately and promptly. (Amin Ul Haq, 2018) has demonstrated a machine-learning-based diagnostic system for reliably predicting heart disease. Utilizing a dataset including information about heart disease, this system would be able to diagnose heart failure. In addition to seven different classifier evaluation metrics, they utilized cross-method validation, seven distinct machine learning approaches, and three feature selection procedures. When evaluating performance, classification precision, specificity, sensitivity, and execution time were considered. Seven machine learning methods were implemented, including logistic regression, artificial neural networks, the Support vector machine approach, Naive Bayes, and random forest. The system was validated using the K-Fold cross validation approach, and its features were selected using the mRMR, Relief, and LASSO feature selection methods. With an accuracy of 89%, the logistic regression model with 10-fold cross validation delivered the best results. Throughout the duration of the study, the Cleveland heart disease dataset was applied to distinguish between healthy and unwell subjects. Utilizing qualities that were irrelevant to the situation at hand led to a loss in the system's overall performance and an increase in the computing time required. In order to further enhance the performance of the predictive machine learning model, a more effective feature selection technique and optimization strategies can be implemented.

Cardiovascular disorders are the leading cause of death worldwide, and in the last few decades, they have grown into a potentially lethal condition. In the study conducted by (V.V.Ramalingam, 2018) 4, machine learning methods including Naive Bayes (NB), SVM, KNN, and DT were employed. In addition, correlation-based feature selection, a dimensionality reduction technique, was employed to extract features and choose which features to employ. The following are the outcomes generated by this ML Model: Using the Cleveland Dataset, RF was able to achieve an accuracy of 91.6% by employing NB, which provided an accuracy of 84.15 percent, SVM, which provided an accuracy of 85.7 percent, KNN, which provided an accuracy of 83 percent, and the decision tree, which provided an accuracy of 82.1 percent.

Due to the vital importance of the heart to human life, the diagnosis of cardiac disease demands exceptional accuracy and precision. (Singh, 2020) Using K-nearest neighbor (KNN), decision tree (DT), and support vector approaches, this study proposes a machine learning model capable of predicting heart illness (SVM). To train and test these algorithms, a dataset from the UCI repository is utilized. To implement the machine learning model, Anaconda and Python are deployed as programming languages. Included in the employed strategy are the following phases: 1) Data collection, of which 37% was utilized for the system testing dataset and 73% was utilized for the training dataset. 2) Age, gender, the presence or absence of chest pain, cholesterol levels, fasting blood sugar, resting time, and the number of major blood arteries were taken into account. 3) The first data processing; 4) The distribution of the data across the target classes; and 5) The histogram of the attributes. Using this model, the results were 83% when utilizing SVM, 79% when utilizing DT, and 87% when utilizing KNN.

A Review on Diabetes Prediction

Diabetes is also regarded as one of the most chronic diseases since it causes an increase in the amount of blood sugar in the human body; if it is not diagnosed, it can lead to a number of issues. The research conducted by (Deepti Sisodia, 2018) presents a machine learning model that can predict whether a patient will develop diabetes or not. This model employs three distinct classification techniques for machine learning: Support vector machine, Decision tree, and Naive Bayes. Using the Pima Indian Diabetes Database available from the UCI ML repository, the model developed using these approaches is trained and evaluated. This model was evaluated based on criteria including precision, accuracy, the F-

measure, and recall. The collected data suggested that Naive Bayes surpassed the competition with an accuracy of 76.30 percent. The development of a system to predict the emergence of new diseases will be part of the future effort.

A comparable work is provided by (R.Aishwarya, 2019), which addresses a machine learning algorithm for accurate diabetes classification. The model employs a single method, namely support vector machine (SVM), as well as principal component analysis (PCA) for preprocessing procedures. The methods employed includes data cleansing, data preprocessing, and the classification of patients. In the algorithm for the support vector machine, linear classification was used, and quadratic equations had to be solved in order to train the model. Due to the fact that only one vector could be trained at a time, Kuhn-Tucker requirements were suggested to preserve adiabatic increments in analytically computed steps. Additionally, the efficiency was enhanced by incremental and decremented orders, and the algorithm searches for the closest point in the opposite class. If the algorithm detects new data or data that is presented in the opposite class, a new margin is created, and this procedure is continued until the data has been thoroughly inspected. Classification is influenced by both the margin of error and the data that is expected to be the newest nearby value. The implementation of SVM classification utilized MATLAB, and the data was preprocessed in anticipation of improved results. In addition, the Principal Component Analysis was employed in order to reduce the dimensions. This resulted in a conclusion as to whether or not diabetes should be categorized. The model's accuracy, sensitivity, and precision were all put to the test.

Diabetes is a chronic condition that can lead to issues with the liver, kidneys, and nerves. These issues could be mitigated with early detection: According to the research presented by Krishnamoorthi (2022), a machine learning model is necessary for making precise forecasts. This model would use an ML-based strategy to design and access diabetes prediction algorithms such as DT, random forest, and Support vector machine. In addition, the research focuses on the development of an intelligent framework for the prediction of diabetes mellitus using machine learning techniques. The model created using this framework was found to be 83 percent accurate, with a very low error rate. In addition to data visualization, preprocessing, classification algorithms, hyper parameter tweaking, comparison analysis, and performance evaluation, a Dataset derived from the Pima diabetes database was utilized in this methodology. After thorough training on the dataset, it was determined that logistic regression worked better and delivered a higher accuracy of 86%. In addition to cancer and Parkinson's disease, the built model may also be capable of predicting other diseases.

A Review on Cancer Prediction

Recently, machine learning has been implemented in the cancer diagnostic and prognostic processes. This latter strategy is particularly promising because it is consistent with the growing trend toward personalized and predictive therapy. The goal of personalized medicine is to treat patients according to their individual features. In actuality, a number of trends have been identified, the most notable of which are an increased reliance on protein biomarkers and microarray data; a strong bias toward prostate and breast cancer applications; and a reliance on somewhat dated technologies such as artificial neural networks rather than a greater number of recently developed or easily interpretable machine learning methods. All of these characteristics add to a high preference for prostate and breast cancer applications. It appears that a considerable proportion of recently published research lacks validation or testing. The application of machine learning algorithms can increase the likelihood of predicting cancer susceptibility, recurrence, and mortality by 15 to 25 percent in studies with improved structure and verification. (Joseph A. Cruz, 2016)

The article by (Yixuan Li1, 2018) proposes a machine learning model for the prediction of cancer in women. The model was trained using two datasets: Breast cancer Coimbra dataset and Cancer in Women in the United States dataset. Breast cancer is one of the invasive cancers and second on the list of cancers that are invasive. The model utilized three distinct metrics, such as the F-measure metric and the AUC value. The procedure of training and testing the data was done more than fifty times, with the following findings for the Breast cancer Coimbra dataset: DT: 68% accuracy; SVM: 71% accuracy; RF: 75% accuracy; LR: 65% accuracy; and NN: 60% accuracy. Similarly, the following results were achieved for the Wisconsin breast cancer dataset: DT: 96% accuracy, SVM: 95% accuracy, RF: 96% accuracy, LR: 93% accuracy, and NN: 100% accuracy. One of the most significant drawbacks of the study was that it only collected information on ten distinct factors. Random forest could also be combined with other algorithms to improve its accuracy and efficiency.

The research conducted by (Konstantina Kourou a, 2014), discusses predicting cancer susceptibility, predicting cancer recurrence, and predicting cancer survival rates. ANN, Bayesian Networks, support vector machine, and decision trees are some of the analytical tools that this study makes use of. The study undertakes an epidemiology study of bladder cancer susceptibility for the purpose of developing classification systems. For cancer susceptibility, Scopus and PubMed were searched in order to find relevant information. The research also included the presentation of a graph that contained a ROC curve and a calibration curve. The ROC curve was used to evaluate the discriminative ability of their model, and the calibration curve was used to compare the measurement of their model to the ideal measurement of predicting breast cancer risk. The accuracy of the outcome was determined to be 84% when using ANN, 71% when using SVM, and 72% when using BN with this model

3. RESEARCH METHODOLOGY

The development framework for this research project was an iterative software development lifecycle model. In this method, the fundamental model is improved through a series of iterative steps until all requirements are met and the program is prepared for operation. Following these steps, a straightforward and basic implementation of a relatively limited number of software requirements is carried out. The iterative software development life cycle (SDLC) model that was utilized throughout the construction of the system is briefly detailed in the subsequent subsections.
An Examination of the Needs and Preferences During this stage, an investigation of the necessary components of the system is carried out. This process will ultimately result in a document referred to as the "System Requirements Specification," or SRS for short.
Planning Stage At this point in the process, the SRS is morphed into a design for the system. Develop context diagrams, class diagrams, use case diagrams, and DFD-ED diagrams.
During the Coding Phase During this stage, the design is programmed in line with the requirements of a healthcare application, and by the time the method is complete, a system that is operational has been produced.
Testing Period In this stage, a test list is utilized to conduct testing and make suggestions for improving the system that has been produced. Additionally, the program is modified, and the software is continuously run until a reliable system is established. provided in the form of a chance increase.

4. DESIGN AND IMPLEMENTATION SPECIFICATIONS

This chapter details the entire design and implementation process, beginning with the methods for setting up the implementation and continuing through its deployment, testing, and provision of statistical data. This presents a synopsis of the benefits and drawbacks of the project, both of which have the potential to be improved upon and expanded. The process of development consists of two primary elements, both of which are linked to two primary software tools.
The delivery of the project will be handled by Amazon Web Services, and Git will continue to be incorporated. As can be observed, the entire development process of the project is not very complicated; however, it has been developed with the scalability of the software development team, cost savings, and ease of maintenance in mind for the healthcare industry.

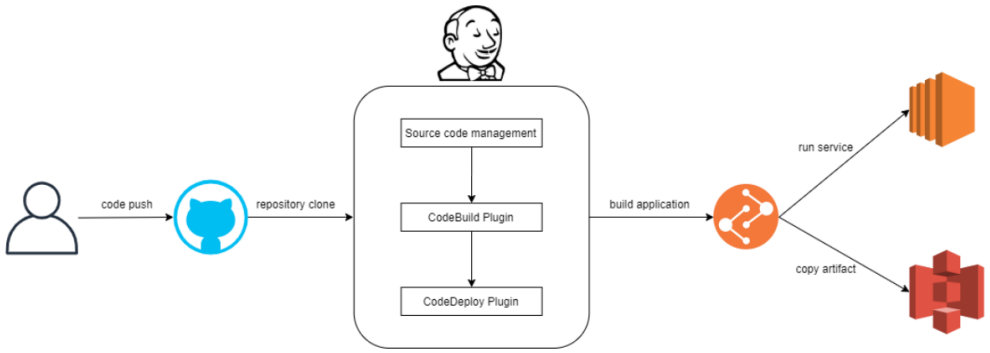

Figure 4. The Block diagram shows the connectivity between Git and AWS Cloud

This report suggests the implementation of a cloud and edge computing system for healthcare systems. This system would make use of AWS, CI/CD with Git, Python (Django) for the development of web applications, and machine learning algorithms for the prediction of diseases and the offloading of tasks to the cloud.

The approach that has been provided would give healthcare systems an effective and economical method of processing data, predicting ailments, and offloading tasks to the cloud. In addition to providing a platform for developing and distributing applications, AWS will also provide services for continuous integration and continuous deployment (CI/CD), such as EC2, S3, and Elastic Beanstalk. Python (Django) will be used for the development, and machine learning methods will be used to anticipate diseases and offload activities to the cloud (Setting up a CI/CD pipeline by connecting Jenkins with AWS Code Build and AWS Code Deploy). Python (Django) will be used for the development.

Configuring Amazon Web Services, starting with the configuration of the S3 bucket. S3 buckets allow users to store up to 5 gigabytes of data at no additional cost, making them an excellent starting point for users. The configuration of an S3 bucket is a straightforward process, and the vast majority of the default settings can be kept. However, certain information, like as the name of the bucket, its location, and the settings for public access, needs to be given. In spite of the fact that AWS will provide a warning message when public access is chosen, doing so is essential for making any future environment preparations. The bucket will be visible in the S3 console once it has been created, which is a far faster process than activating other services. In addition, a policy setup will need to be finished if the S3 bucket is going to be functional. This policy manages access to the bucket's resources through the use of a language that is based on JSON

Aws Code deploy:

AWS Code Deploy is a completely managed deployment solution that automates software deployments to a range of computer services including Amazon EC2, AWS Fargate, AWS Lambda, and our own on-premises servers. This service is offered by Amazon Web Services (AWS).
Code Deploy simplifies the process of swiftly releasing new features, assists in preventing downtime during application deployment, and manages the complexities of updating your applications.
Control from a central location, The AWS management console is used in conjunction with AWS Code Deploy to make it simple for you to launch application deployments and monitor their progress. During the process of software deployment, AWS Code Deploy helps to maximize the availability of your application so that you have the least amount of downtime possible.

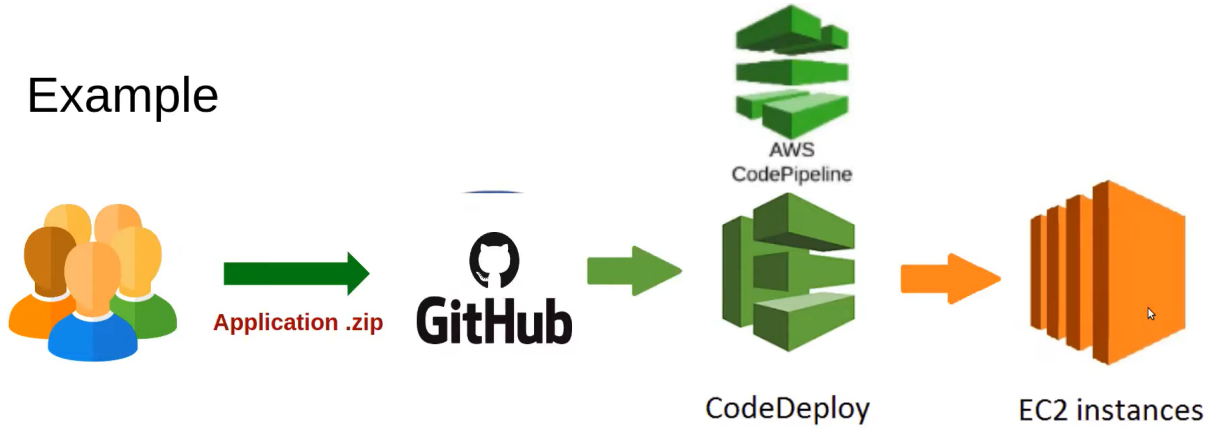

Figure 5. The figure shows the flow diagram of CI/CD Pipeline

Access control lists (ACLs), bucket policies, access point policies, or all three can be used in AWS S3 to provide public access to buckets and objects. Turning on Block all public access will prevent unauthorized users from accessing the contents of this bucket and the objects contained within it. These settings are only applicable to this bucket and the access points associated with it. AWS suggests that you switch to the Block all public access setting; however, before implementing any of these settings, you should make sure that our applications will continue to function normally even if

public access is disabled. In the event that we require some degree of public access to this bucket, or the objects contained within it, we are able to tailor each of the parameters listed below to meet the requirements of your particular storage use cases.
S3 bucket settings do not come with a website hosting option selected by default. These settings should be changed to include a bucket home page as well as an error page directory (e.g., index.html and error.html). The most essential function of an S3 bucket is to serve as a repository for data; hence, it will soon be possible to use Jenkins to automate the process of transferring data from a source to the bucket. Due to the fact that the Jenkins task was not available, it was necessary to complete this work manually by dragging and dropping files into the upload section. It is suggested to set up CloudFront Service in conjunction with an S3 bucket in order to deliver and secure the material. Doing so will ensure safety, improve performance, and keep costs under control. CloudFront plays an essential part in the delivery of data to consumers, as well as in the encryption of communications using a unique SSL certificate and the automatic provision of DDoS protection using AWS Shield Standard. Integrating the S3 service into its design is crucial.

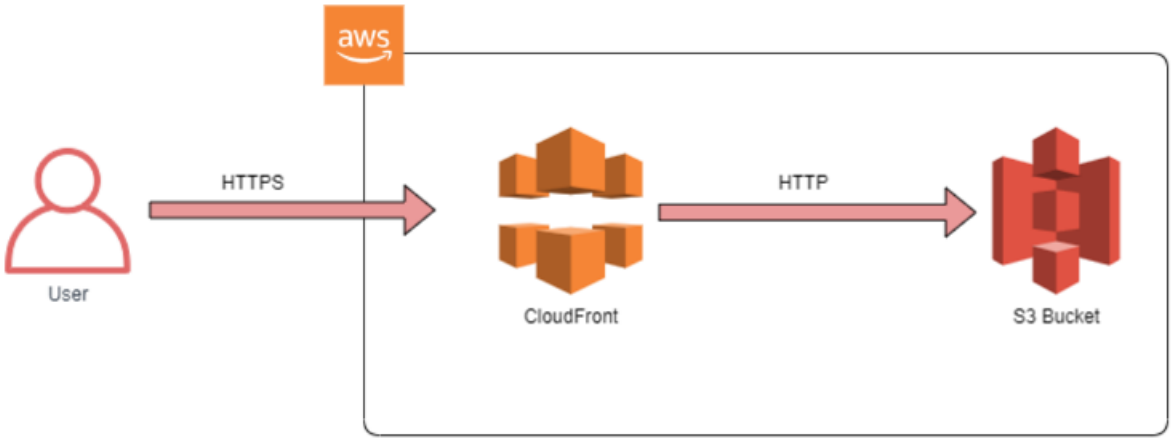

Figure 6. The figure shows the AWS S3 service diagram

CloudFront is essential for the development of a web distribution type that can simultaneously serve static and dynamic content. The Origin Domain Name field on the Create Distribution page must be filled in with the name of the Amazon S3 bucket that was created earlier; doing so will cause the distribution ID to be automatically inserted into the Origin ID field. The rest of the settings fields, such as Origin Path, Origin Customer Headers, and Restrict Bucket Access, are all optional and can have their values left as the defaults. It is also possible to keep the Default Cache Behavior Settings field in its default state for the time being the distribution behavior can be altered in the future when a personalized SSL certificate for an HTTPS connection is made available. The Distribution Settings part of the procedure is the most vital part of the process. This part of the procedure provides inputs for Alternative Domain Names (CNAMEs) and options for SSL certificates. People have the option of entering their own personalized domain name, which may be made with a DNS Service acting as a CNAME record or by using an AWS origin that ends with ". cloudfront.py." In addition, this section can be set to obtain the same Default Root Object as S3, and as a result, it can target the appropriate location on the bucket.

Elastic Load Balancing, A public DNS service offers enhanced request routing capabilities for the delivery of current application architectures like microservices and containers. These features are essential for modern application delivery. Working at the level of individual requests (Layer 7), it uses the information contained in the request to direct traffic within Amazon Virtual Private Cloud (Amazon VPC) to healthy targets. This ensures that incoming application traffic is automatically and uniformly distributed across numerous targets inside a single Availability Zone, which enables a service that is both highly dependable and scalable. Targets can include things like Amazon EC2 instances. A load balancer is guaranteed to be available 99.99% of the time under the terms of the Service Level Agreement (SLA) offered by Amazon Elastic Load Balancing.

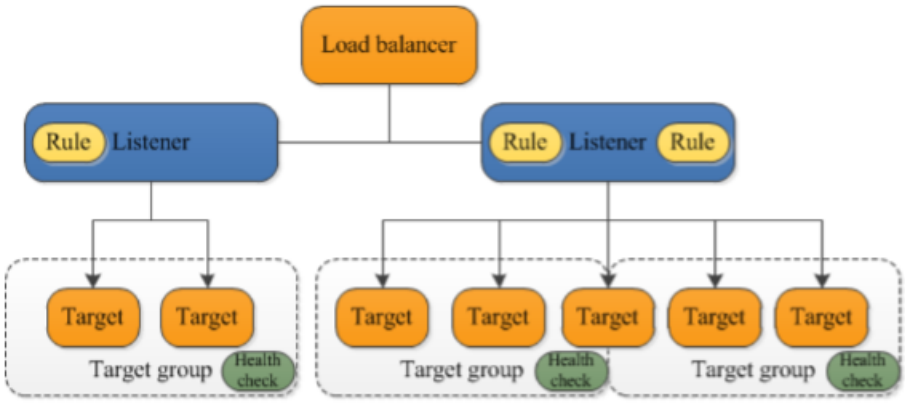

Figure 7. Shows a High-Level block diagram of a Load Balancer

Installing SSL certificate and sub domain

Using AWS Certificate Manager and Route 53, it's easy to set up an SSL certificate and a subdomain. People can request an SSL certificate with just a few clicks in the Certificate Manager service. This makes it easy to set up an SSL certificate. Validation of the certificate is also made easier. Based on the information in the certificate, the service will automatically create a CNAME record in the Route 53 service, or people can choose to create the record themselves. The Route 53 service makes it easy to set up subdomains, and an alias record can be used to direct a subdomain to an AWS resource.

Description of a heart disease dataset

The dataset contains information regarding age, sex, chest pain, cholesterol, blood pressure at rest (trestbps), serum cholesterol in mg/dl (chol), fasting blood glucose >120 mg/dl (fbs), resting ECG result (restecg), maximal heart rate reached (Thalach), exercise-induced angina pectoris (exang), exercise-induced ST depression relative to resting (old peak), slope of highest exercise ST segment (slope), and the number of major vessels (0- (thallium stress results). The dataset has 1025 different date samples in total. In spite of the fact that the goal data are projections that are reliant on the input values of a particular dataset, the data that are presented above are input data that are distinct from one another. The outputs for the target values consist of the numbers 0 and 1, which indicate whether or not the cardiac disease has been found.

Description of a Diabetes dataset –

This data set contains information pertaining to diabetes reports, including details on blood pressure, skin thickness, insulin, body mass index, and the function of the diabetic hereditary gene. The dataset has 1025 different date samples in total. The data that were just discussed are not reliant on one another in any way; instead, the outcomes are determined by the input values of the dataset. The values 0 and 1 are included in the output of the result value. This reveals whether or not there are any problems associated with diabetes.

Random Forest Classifier

Both regression and classification are accomplished with the help of a method known as Random Forest, which belongs to the realm of supervised learning. Nevertheless, classification problems remain its principal application. Random forest algorithms build a decision tree out of data samples, derive predictions from each, and then vote on which one provides the most accurate result. This is known as an ensemble technique, and because it averages the results, it is superior to a single decision tree because it reduces the likelihood of overfitting occurring. The phases of operation for the random forest method are detailed below in the following list.

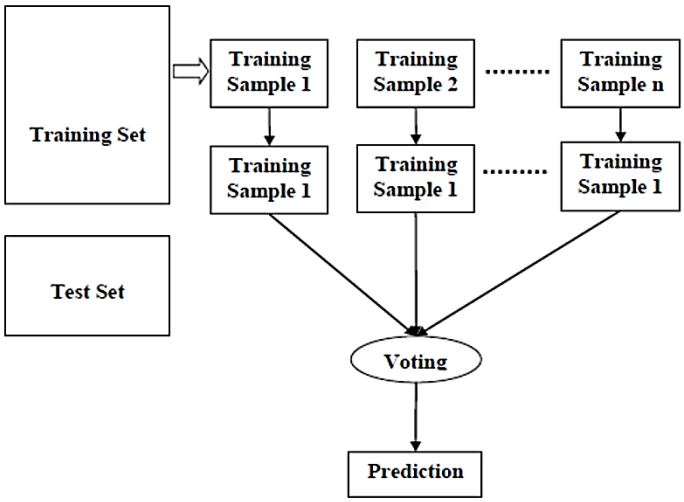

Figure 8. The figure shows the flow diagram of the Random Forest Classifier

First, we select a sample at random from an already existing dataset so that we may get started.
In the second step of the method, a decision tree is constructed for each of the samples. The predicted outcome can then be derived from each of the decision trees.
Step 3 consists of casting votes for the various outcomes that were anticipated.
Step 4: As the final step, we determined which of the possible outcomes of the prediction had the most votes and made that our final prediction outcome. The following diagram outlines the process that the algorithm goes through.

5. RESULTS AND DISCUSSION

Processing the heart disease dataset-

Following an analysis of the dataset, we concluded that, in order to properly train the machine learning models, it was necessary to first normalize all of the values and convert some category variables into dummy variables. To get things rolling, we will use the acquire dummies method included in the pandas' package to create dummy columns for the category variables.
K-Nearest Neighbors is the name of the algorithm (KNN)

K-Nearest Neighbor is one of the more straightforward machine learning algorithms, and it uses the supervised learning approach as its foundation.

• The K-NN method assigns new cases to the category that most closely resembles the available categories based on similarities between new instances/data and existing cases. This is done by comparing the new cases to the existing cases and looking for patterns.

• The K-NN algorithm organizes fresh data points into categories according to their degree of similarity and stores all of the data that is currently available. This indicates that new data can be swiftly categorized using the K-NN method whenever it is generated, which is a significant benefit.

• Although it is possible to apply K-NN algorithms to classification and regression problems, the most common use for these algorithms is in the context of classification problems. An algorithm known as K-NN was utilized in order to make projections regarding heart disease. It is possible to describe the operation of K-NN by using the algorithm shown below:

First, determine the number of neighbors that you desire K.

In the second phase, you will compute the Euclidean distance between each pair of K neighbors.

Find the K neighbors who are geographically closest to you based on the Euclidean distance that was determined in step 3.

The fourth stage in analyzing this set of k neighbors is to count the number of data points that fall under each category.

The fifth stage consists of adding new data points to the category that has the most adjacent categories. At this point, the model has been prepared. It is necessary for us to include a fresh data point in the selected category. Take into consideration the illustration in fig 9.

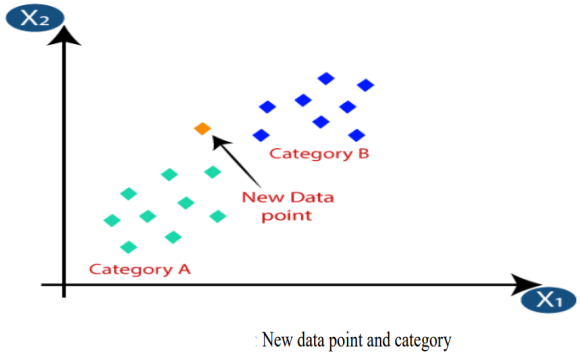

Figure 9. Graphical representation of the data points

Because we first select the number of neighbors, i have considered that k should equal 5. After that, the Euclidean distance between each pair of data points is figured out. The "Euclidean distance" between two points is the term used to describe the distance between them. It is possible to compute it using the graph that will be shown in fig 10.

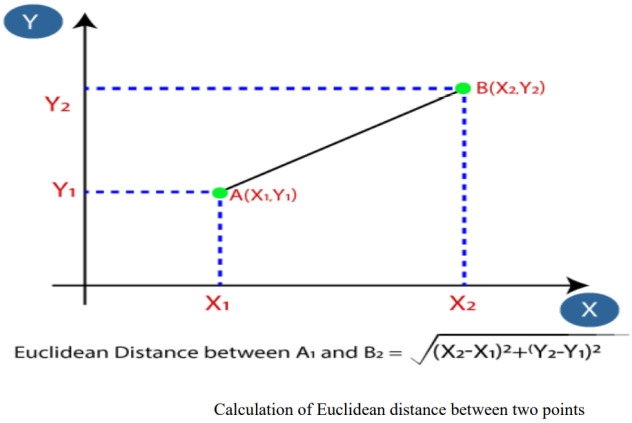

Figure 10. Graphical representation of the Euclidean Distance

After that, we determined the Euclidean distance between each pair of points and located the two nearest neighbors for category B and the three nearest neighbors for category A respectively. Consider the following illustration in your thinking.

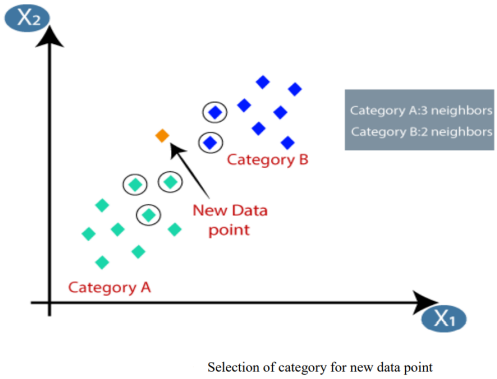

Figure 11. Graphical representation of new data point

Because category A includes the three data points that are geographically closest to this new data point, it is clear that this new data point must also be included. When determining the value of K for the K-NN algorithm, the following considerations need to be considered.

You will need to try out a number of different possibilities for the letter "K" because there is no foolproof way to decide which one is the best choice. The number 5 is the most accurate representation of the letter K. Values of K that are extremely low, such as K=1 and K=2, might be noisy and lead to outliers in the model.

Large K values, despite the fact that they have a number of benefits, can also bring about a number of problems. The dataset that we used to determine the accuracy of the K-NN algorithm for predicting heart disease was split in two. To put it another way, 75% of the data were used for the training size, whereas only 25% of the data were used for the test size. The accuracy rating of the algorithm increases to 91.77% after training, however it drops to 81.82% when it is tested on newly collected data.

```
Train Result:
================================================
Accuracy Score: 91.77%

________________________________________________
CLASSIFICATION REPORT:
                    0      1  accuracy  macro avg  weighted avg
precision        0.91   0.92      0.92       0.92          0.92
recall           0.91   0.92      0.92       0.92          0.92
f1-score         0.91   0.92      0.92       0.92          0.92
support        340.00 377.00      0.92     717.00        717.00

________________________________________________
Confusion Matrix:
 [[310  30]
 [ 29 348]]

Test Result:
================================================
Accuracy Score: 81.82%

________________________________________________
CLASSIFICATION REPORT:
                    0      1  accuracy  macro avg  weighted avg
precision        0.86   0.78      0.82       0.82          0.82
recall           0.77   0.87      0.82       0.82          0.82
f1-score         0.81   0.82      0.82       0.82          0.82
support        159.00 149.00      0.82     308.00        308.00

________________________________________________
Confusion Matrix:
 [[123  36]
 [ 20 129]]
```

Figure 11. shows the accuracy results

Analysis of the diabetes dataset –

The data shown in the image below does not contain any null values and is one of the numerous variables in our dataset that are represented by integer and floating-point values respectively. As was just discussed, the record corresponding to the value of this outcome contains both zeros and ones. The number of values with a 1 corresponds to 684, whereas the total number of values with a 0 corresponds to 1316; this demonstrates the quantity of typical diabetes report data. a number that indicates information regarding diabetes. The information presented above is shown with visuals in the image that can be seen below in the fig 12.

```
[5]: diabetes_data.info(verbose=True)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               2000 non-null   int64
 1   Glucose                   2000 non-null   int64
 2   BloodPressure             2000 non-null   int64
 3   SkinThickness             2000 non-null   int64
 4   Insulin                   2000 non-null   int64
 5   BMI                       2000 non-null   float64
 6   DiabetesPedigreeFunction  2000 non-null   float64
 7   Age                       2000 non-null   int64
 8   Outcome                   2000 non-null   int64
dtypes: float64(2), int64(7)
memory usage: 140.8 KB
```

Figure 12. Shows the elements in the Diabetes dataset

The following graph illustrates the correlation between age and the highest rate of diabetes that a person can achieve in their lifetime. It has been demonstrated that the largest prevalence of diabetes occurs between the ages of 40 and 70, and it is believed that this is also the age range in which diabetes produces the most serious consequences. This is illustrated in the graphic that can be found below in fig 13:

Figure 13. Occurrence of diabetes between age 40-70

There is not a single data point within the file that lacks a value for any of the datasets that were previously discussed. The figure that follows displays, for each individual parameter that is included in the datasets, the total number of counts that can be found for that particular parameter, as shown in the fig 14.

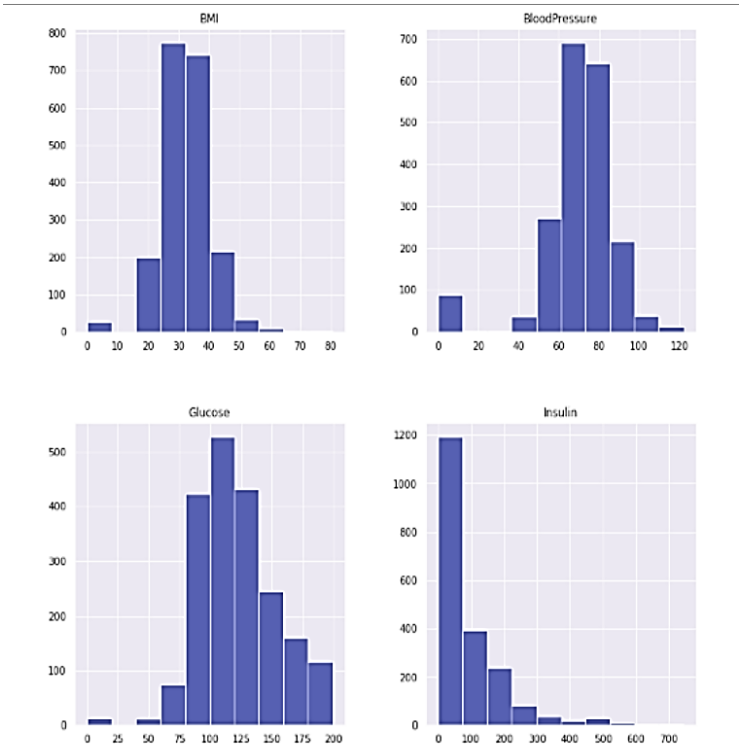

Figure 14. Shows the graph of individual parameter results

K-NN Accuracy for Diabetic Disease Prediction

The dataset was cut to create two distinct parts. To put it another way, 75% of the data were used for the training size, whereas only 25% of the data were used for the test size. The accuracy rating of the algorithm increases to 88.40% after training but drops to 81.20% when it is tested on newly collected data. The outcomes of the draw using the confusion matrix There are 1.887 real positives in the dataset, whereas there are 439 genuine negatives. The number of true positives in the dataset is 1.887. There are a total of 95 errors, with 2.79 mistakes. 3. There have been discovered to be 95 false positives. To put it another way, the algorithm made a positive prediction about the outcome, but the outcome itself is deceptive. 4. The computer software generated 79 incorrectly pessimistic forecasts, all of which turned out to be wrong. Because of this, the accuracy of the prediction or aggregation will be ensured if we compute the precision. In other words, true positive, false negative, false positive, and true negative findings, respectively, are denoted by the abbreviations TP, FN, FP, and TN, respectively.

```
Train Result:
================================================
Accuracy Score: 88.40%
------------------------------------------------
CLASSIFICATION REPORT:
                    0           1    accuracy     macro avg  weighted avg
precision    0.918219    0.822097    0.884      0.870158      0.885025
recall       0.903259    0.847490    0.884      0.875375      0.884000
f1-score     0.910678    0.834601    0.884      0.872639      0.884406
support    982.000000  518.000000    0.884   1500.000000   1500.000000
------------------------------------------------
Confusion Matrix:
 [[887  95]
 [ 79 439]]

Test Result:
================================================
Accuracy Score: 81.20%
------------------------------------------------
CLASSIFICATION REPORT:
                    0           1    accuracy    macro avg  weighted avg
precision    0.846821    0.733766    0.812     0.790294      0.809287
recall       0.877246    0.680723    0.812     0.778984      0.812000
f1-score     0.861765    0.706250    0.812     0.784007      0.810134
support    334.000000  166.000000    0.812   500.000000    500.000000
------------------------------------------------
Confusion Matrix:
 [[293  41]
 [ 53 113]]
```

Figure 15. Shows the accuracy results of the K-NN Classifier

Accuracy of the CNN Algorithm

The model has an accuracy rating of 91.02% when being tested. Taking everything into consideration, this is of very high quality. utilized data size. In addition to this, the accuracy of the model's training is 98.23%.

Accuracy of the random Forest Algorithm

A random forest method was used after the dataset was initially segmented into two parts: 30 percent for the testing phase and 70 percent for the training phase. The accuracy score of the algorithm was 94.34% after it had been trained, but it dropped to 93.70% when it was evaluated on new data.

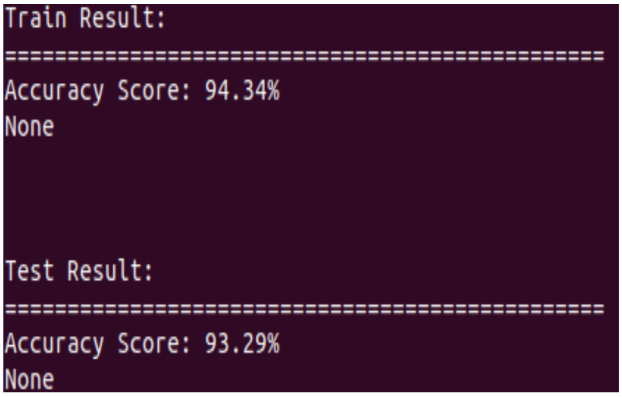

Figure 16. Shows the accuracy results of CNN Classifier

Security for Web Data

When developing a web-based application, the most significant challenge is ensuring the application's safety. Because our smartphone has access to all of the information on the web while using this program, the information itself must be secure. When we talk about data security, the first thing we take into consideration is physical security, also known as the protection of data from thieves, natural disasters, and other types of physical damage. It is possible that the data will be used inappropriately if they are stolen, in addition to being damaged. Physical security can be provided by guards and administrative staff, but the data that is stored on the internet cannot be protected by them. This should be our primary concern. To protect one's data while it is being stored on the web, one may take the following steps:

• Protecting data from being injected with SQL. There have been multiple high-profile attacks on online programs, and one of them, a successful SQL injection attack that stole credentials, is suspected of being connected to these attacks. The method of code injection known as SQL injection could end up destroying our database. SQL injection is one of the most common methods used for hacking websites online. The act of inserting malicious code into SQL queries through the input on a web page is referred to as SQL injection. When writing code, you can circumvent this problem by using prepared statements.

• Data on the web needs to be protected from both malicious hackers and unauthorized users. The firewall that is available on our system can be utilized to protect data and prevent the introduction of unwanted files that might compromise it or lead to the loss of data. Using MD5 or any of the other standard hashing methods may result in a more secure protection for our login information, which would prevent unauthorized access.

6. CONCLUSION AND FUTURE WORK

Convolution neural networks, random forest algorithm, the K-NN algorithm, and a number of other algorithmic procedures have all made significant contributions to the development of the E-Health Care system, which has been successful. This web-based application will be of great utility because it is not only efficient but also an excellent method for lowering the rate of medical errors and enhancing clinical judgment. This system establishes an environment that facilitates an easier preliminary assessment of a patient's health by providing the patient with the ability to check for diseases based on symptoms and a doctor's recommendation to consult with a specialist when necessary. The system will be helpful for setting up an appointment for a consultation between a patient and a doctor. In addition to that, it makes it possible for the doctor to send online subscriptions to the patient's email address based on what the patient requires. The requirements of emerging tendencies and technologies, in addition to the requirements of the centralized control of all of our systems via ICT, have resulted in an increased demand for electronic health care applications. This application will, in the not-too-distant future, ensure that everyone has access to health care that is both affordable and of a high quality, while also accelerating the much-required reform of our health care systems. This research presents an overview of using cloud and edge computing for healthcare systems, with a focus on using AWS for CI/CD with Python (Django) for development, and machine learning algorithms for disease prediction and task offloading to the cloud. This implementation would allow healthcare systems to improve their data processing capabilities and better detect, diagnose, and treat diseases. This research can be expanded based on the level of automation that users expect, and there is still room for improvement in the future for the upcoming expansion feature of the system.

REFERENCES

1. Wang, S., Xu, J., Zhang, N. and Liu, Y., 2018. A survey on service migration in mobile edge computing. *IEEE Access*, *6*, pp.23511-23528.
2. Zhao, X., Zhao, L. and Liang, K., 2016, July. An energy consumption oriented offloading algorithm for fog computing. In *International conference on heterogeneous networking for quality, reliability, security, and robustness* (pp. 293-301). Springer, Cham.
3. Mohapatra, S. and Rekha, K.S., 2012. Sensor-cloud: a hybrid framework for remote patient monitoring. *International Journal of Computer Applications*, *55*(2).
4. Chen, M., Qian, Y., Chen, J., Hwang, K., Mao, S. and Hu, L., 2016. Privacy protection and intrusion avoidance for cloudlet-based medical data sharing. *IEEE transactions on Cloud computing*.
5. Hossain, M.S. and Muhammad, G., 2016. Cloud-assisted industrial internet of things (iiot)–enabled framework for health monitoring. *Computer Networks*, *101*, pp.192-202.
6. Haq, A.U., Li, J.P., Memon, M.H., Nazir, S. and Sun, R., 2018. A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms. *Mobile Information Systems*, *2018*.
7. Ramalingam, V.V., Dandapath, A. and Raja, M.K., 2018. Heart disease prediction using machine learning techniques: a survey. *International Journal of Engineering & Technology*, *7*(2.8), pp.684-687.
8. Singh, A. and Kumar, R., 2020, February. Heart disease prediction using machine learning algorithms. In *2020 international conference on electrical and electronics engineering (ICE3)* (pp. 452-457). IEEE.
9. Aishwarya, R. and Gayathri, P., 2013. A method for classification using machine learning technique for diabetes.
10. Sisodia, D. and Sisodia, D.S., 2018. Prediction of diabetes using classification algorithms. *Procedia computer science*, *132*, pp.1578-1585.
11. Krishnamoorthi, R., Joshi, S., Almarzouki, H.Z., Shukla, P.K., Rizwan, A., Kalpana, C. and Tiwari, B., 2022. A novel diabetes healthcare disease prediction framework using machine learning techniques. *Journal of Healthcare Engineering*, *2022*.
12. Cruz, J.A. and Wishart, D.S., 2006. Applications of machine learning in cancer prediction and prognosis. *Cancer informatics*, *2*, p.117693510600200030.
13. Li, Y. and Chen, Z., 2018. Performance evaluation of machine learning methods for breast cancer prediction. *Appl Comput Math*, *7*(4), pp.212-216.
14. Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V. and Fotiadis, D.I., 2015. Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, *13*, pp.8-17.

15. Chen, M., Qian, Y., Chen, J., Hwang, K., Mao, S. and Hu, L., 2016. Privacy protection and intrusion avoidance for cloudlet-based medical data sharing. *IEEE transactions on Cloud computing*.
16. Sultan, N., 2014. Making use of cloud computing for healthcare provision: Opportunities and challenges. *International Journal of Information Management*, *34*(2), pp.177-184.
17. Ahuja, S.P., Mani, S. and Zambrano, J., 2012. A survey of the state of cloud computing in healthcare. *Network and Communication Technologies*, *1*(2), p.12.
18. Al-Marsy, A., Chaudhary, P. and Rodger, J.A., 2021. A model for examining challenges and opportunities in use of cloud computing for health information systems. *Applied System Innovation*, *4*(1), p.15.
19. Aziz, H.A. and Guled, A., 2016. Cloud computing and healthcare services
20. May, R. and Denecke, K., 2022. Security, privacy, and healthcare-related conversational agents: a scoping review. *Informatics for Health and Social Care*, *47*(2), pp.194-210.