Machines and Algorithms
http://www.knovell.org/mna

Research Article

A Deep Learning Based Approach to Breast Cancer Detection

Muhammad Nadeem[1, *] and Muhammad Nabeel Asghar[1]

[1]Department of Computer Science, Bahuddin Zakariya University, Multan, 60000, Pakistan

[*]Corresponding Author: Muhammad Nadeem. Email: mud.nadeem333@gmail.com

Received: 03 May 2023; Revised: 07 June 2023; Accepted: 26 July 2023; Published: 16 August 2023

AID: 002-02-000021

Abstract: Breast cancer is the most frequent malignancy in women worldwide. Lack of understanding and late tumor diagnosis increase women's mortality. Early discovery, therapy, and periodic check-ups may reduce mortality. Breast tissue cells proliferate too quickly, causing cancer. Numerous researchers calculate breast cancer tumor accuracy and prediction using different machine learning models on different datasets. Researchers employ CNN and other deep learning algorithms. This study classifies benign and malignant cancers using inceptionv3 deep learning model and a convolutional neural network using convolution layers. The DDSM Mammography dataset comprises 11170 pictures, while the BreakHis dataset has 7909 images. This study trains the CNN model inception V3. A PCA-based logistic regression classifier outperformed other machine learning algorithms. This study uses transfer learning on a pre-trained proposed model, Inception V3, which has a testing accuracy of 96% on the DDSM dataset and on the WCBD dataset the accuracy of 96.24%. The third dataset, the Breast Cancer Histopathological Database (BreakHis), has 7909 microscopic pictures, and the CNN model had 98.8% accuracy, the best of all datasets in this study. Cross-validation of accuracy, precision, recall, and f1 score on deep learning approaches improves results.

Keywords: Breast cancer; malignancy; mortality; early discovery; CNN; deep learning;

1. Introduction

Breast cancer is caused by the hasty progress of cells in the breast tissue. These tissues are tumors, which can be benign or malignant. Noncancerous is benign, while cancerous is malignant. Malignant is more dangerous because it spreads quickly to other parts of the body, resulting in death. Research states that “2.3” million new breast cancer patients are diagnosed globally and over “650,000” die from it [1]. Globally, one woman gets breast cancer every 20 seconds and dies every 5 minutes. Over “170,000” new cancer cases are expected in Pakistan this year. Before it spreads and kills, breast cancer must be stopped. Machine learning (ML) is crucial to medical image categorization. In recent years, ML approaches have been developed for manual and automatic illness identification. ML algorithms are advancing, including deep learning. Deep learning (DL) is a smarter, more advanced machine learning area. DL approaches significantly impact medical image classification models [2].

Images are classified using mammography, magnetic resonance imaging (MRI), ultrasound, and biopsy in medical science. A biopsy determines if a bodily component is malignant. Different sources said 80% of women are breast cancer-free after a biopsy. A mammogram diagnoses breast cancer via an X-ray. If the

X-ray diagnoses the afflicted area, the doctor proposes evaluating the breast tissue. When a suspicious breast area is found, the doctor orders an ultrasound. If the ultrasound does not show considerable tissue, the doctor may recommend an MRI to better see the breast [3]. Breast cancer tumors are detected by taking numerous breast MRI pictures and syndicating them on the computer.

Breast cancer is diagnosed via screening, when physicians or nurses look for tumors using mammography and other imaging methods. Early cancer detection is possible with screening [6]. Mammography is the best early breast cancer diagnostic method due to its low cost and quickness. In mammography, anomalies like masses and calcifications are analyzed to diagnose breast cancer. Physicians estimate mammography accuracy at over 90%. Doctors may miss 11%–16% of breast cancer cases. Cross-validation requires two doctors to evaluate the same mammogram at separate times [7]. Although cross-validation improves breast cancer diagnosis to 14% with single checking, this procedure is time-consuming and costly. Computer-aided detection reduces this cost. In medical analysis, computers can be used. Databases, machine learning, image processing, and data analysis are used for this detection [8].

How might the computer-aided design (CAD) system reduce mammography breast cancer false positives?

Classifying pictures with image processing and computer vision reduces early breast cancer false-positives. Breast cancer mammography diagnosis is supported by these two machine learning methods. Edge detection, noise reduction, picture pre-processing, and region of interest are used to detect breast abnormalities in mammograms. After picture pre-processing, find characteristics to develop an algorithm to accurately diagnose breast cancer. These methods' precision is still a challenge.

This research presents a mammogram-based breast cancer detection method. The technique has two main aspects. Image processing is utilized to extract features from the DDSM and BreakHis datasets in the first portion. InceptionV3, a neural network model, predicts model accuracy using extracted features. On the WCBD dataset, logistic regression and the other three machine learning algorithms are trained and assessed in Part 2. This research aims to improve breast cancer diagnostic accuracy by integrating image processing and supervised machine learning classifiers like logistic regression into the new PCA model. This research also aims to eliminate the false positive probability from the breast cancer detection confusion matrix on WCBD and DDSM.

This section provides an introductory summary of breast cancer, including its detection methods and research objectives. Section 2 provides a comprehensive assessment of recent relevant studies. Section 3 presents a thorough and inclusive examination of the proposed methodology. In the fourth section, we learn about the trials that were run to assess the efficacy of deep learning models and machine learning algorithms in breast cancer diagnosis using the DDSM dataset. The fifth section provides a comprehensive account of the findings and recommendations for future research.

2. Related Work

S.V. Sree reviews breast cancer studies in this paper. The CNN deep-learning algorithm detected breast cancer [14]. Input, output, and hidden layers make up a deep neural network. The model's intermediate layer is hidden. CNN comprises four layers: convolution, max pooling, Relu, and SoftMax. X-ray mammograms are among the many diagnostic tests. This article concludes that deep learning performs better on image data.

B. Jaafar [15] utilized deep learning to interpret mammogram pictures. Alex Net has 5 convolutional, 3 pooling, and 2 FC layers. Resnet is the latest deep learning algorithm with greater shortcuts and batch normalization. Auto-encoders and decoders are used to segment and detect tasks in this paper. Classification using CNN. Deep learning expanded mammography analysis. Mammography and CNN breast cancer detection still struggle with massive data.

K. S. Krishna estimated in this research that 2,778,850 people will have breast cancer by 2040. This paper diagnoses breast cancer using machine learning and deep learning. In machine learning, RVM outperforms SVM, and in deep learning, AUC outperforms. This research evaluates three machine learning

models: Random Forest, Naïve Bayes, and KNN, calculating accuracy, precision-recall, and f1 score using the Wisconsin Diagnosis Breast Cancer dataset [17]. KNN excels in accuracy (94%), precision-recall, and f1 score compared to Random Forest (92%), and Naïve Bayes (87%). Using supervised learning could help detect breast cancer early.

N. Khuriwal employed CNN and a deep learning algorithm to diagnose breast cancer early with 98% accuracy on the Mias dataset [18]. Mias has 200 photos and 12 features. The author employed adaptive mean filtering, color-enhancing, and watershed segmentation for pre-processing. The author plans to test on a large image dataset and various cancers like lung "lips," etc.

U.Khasana employed the ultrasound image modality for breast cancer detection, but the picture quality was low; therefore, he applied segmentation and the watershed transform method to improve image quality and get 88.6% accuracy with about 11% error [19].

This research recommends the automatic Diverse Features-based Breast Cancer Detection (DFEBCD) algorithm to classify mammograms as normal or benign [20]. Emotional Learning-inspired Ensemble Classifier (“eliec”) and Support Vector Machine (SVM) on the IRMA mammography dataset Diverse Features-based Breast Cancer Detection (dfebcd) were used to test CNN and other classifiers. CNN was good for three people, but hybrid and dynamic characteristics made “eliec” the best classifier.

ML approaches for breast cancer detection are discussed in this research. Mr. Rathi suggested a mixed strategy. They compared MRMR feature selection with four machine learning classifiers (Support vector machine, function tree, Naïve Bays, and end meta) based on characteristics such as absolute error, accuracy, Kapa statics, specificity, and sensitivity [21]. The author tests these methods using two UCI repository binary and multi-classification datasets. SVM performed better than the other three algorithms in this research [21].

Machine learning is crucial in medicine. A Bharat proposed a breast cancer prediction machine learning system [22]. This research compared the accuracy of four algorithms: k nearest neighbors, support vector machine, decision tree, and naïve bays. K nearest neighbors had the best accuracy of the three algorithms. Multi-SVM can be used for multi-class datasets, although support vector machine techniques only operate for binary classes.

M.Jannesari's fine-tuned deep neural network for four cancers had 99.8% accuracy. On Breakhis, use Resnet V1 50 and V1 152 [23]. The author suggested automated multi-classification breast cancer detection. Convolutional networks for biomedical image segmentation, deep labv3, and U-net are suggested for future research [24].

M.O.F. Goni uses probabilistic neural network (PNN) classifiers and Gaussian mixture models for segmentation. Future neural network classifiers for cancer prediction: benign or malignant probability the author wants cloud-based data for faster access and time savings. These algorithms—Adaptive Mean, GMM, and PNN—help doctors diagnose and cure breast cancer early, saving lives [25].

P. Kathale devised a random forest model to predict breast cancer in normal and cancer patients [26]. The RF model classifies a dataset with 95.8% accuracy after image pre-processing in 6.25 seconds and 3.16 seconds, respectively. GLCM, entropy, and mean image processing features. Future training should use larger datasets for improved accuracy.

M. Kumari presented two breast cancer detection machine learning models in this research. Breast cancer is the most common kind in women and raises the death rate owing to late detection and treatment. KNN and SVM algorithms were applied to the WBCD dataset and yielded 99.28% accuracy. First-stage breast cancer detection with a trained and improved model can save many lives [27]. WBCD data from the UCI repository has 699 values and 11 features, 16 of which are missing. The dataset distributes 65% malignant and 35% benign values. Confusion matrix and cross-validation assessed classifier accuracy. Its accuracy is higher than that of the KNN classifier. Physicians and patients saved time and money using the proposed system. In the future, the author wants a more accurate, cost-effective dataset with more values.

Researchers say more than 2 million women worldwide are diagnosed with breast cancer each year, and the rising death rate causes major health difficulties. Gerald SZE improves classifier accuracy by reviewing Python code for basic algorithms using the Wisconsin breast cancer database. The author chooses the best supervised machine learning algorithm for early breast cancer detection [28]. SVM's linear performance was 97% better than other algorithms, but when dealing with people's lives, a near-perfect model is needed for further investigation.

P.S. Shekar employed the SVM model to manually classify breast cancer histology images as benign or malignant [29]. This article identifies benign and malignant breast masses quickly and precisely. This reduces the risk of bareness and improves survival with several drugs. The paper aimed to create an SVM classifier that can diagnose breast cancer as benign or malignant with 97% accuracy and 95% precision.

C. Singhala introduced deep neural network (DNN) and contrast-limited-based histogram equalization for early breast cancer diagnosis [30]. All MIAS dataset photos were processed using both methods. In experiments, the DNN-based strategy had a lower mean square error (MSE) than the peak signal-to-noise ratio (PSNR), which was high. The author concluded that the DNN-based strategy is the best method for breast cancer detection on the MIAS Database because it performs better.

Z. Wang advocates using computer-aided diagnostics to detect breast cancer early on mammograms. The radiologist still struggles with CAD system accuracy [31]. The CNN learning model can categorize images using deep features and ELM clustering, according to the author. Using an extreme learning machine classifier, the author fused deep density, morphological, and texture data to classify breast tumors as benign or cancerous. The dataset included 400 mammography pictures, 200 of which were benign and 200 malignant. The CNN feature model's ELM specificity, sensitivity, and accuracy are best in a single feature model, suggesting this paper's model is superior. The CNN model with the texture feature in double feature detects breast cancer tumors with the highest specificity, sensitivity, and accuracy.

F. Yilmeez compared dense net-201 with Xception-net for early breast cancer diagnosis [18]. This research evaluates approaches using the breast cancer dataset, which comprises 20748 training images and 5913 testing images. Dense Net-201 has F1 accuracy of 92.24%, while Xception has 92.41%. Both methods have good accuracy and are similar.

To detect benign or malignant cancers, mammography screening is effective. The biggest challenge is identifying benign or malignant patients. Machine learning could improve breast cancer diagnosis. K closest neighbor (KNN) ML algorithm accuracy for early breast cancer diagnosis varies in research publications [32]. S. E. Khorshid said KNN is the easiest ML algorithm to construct and has the best accuracy (99.12%). Using diverse algorithms' accuracy could improve prediction efficiency in the future.

A pre-trained Resnet-50 model and class activation map technique were proposed by Wael E. Fathy to diagnose breast cancer [5]. The suggested method had 82.1% specificity, 99.8% sensitivity, and a 96% AUC. In this paper, a model diagnosed cancer with 93.67% accuracy and 0.122 false positives per image. This study classified pictures as benign or cancerous using the DDSM database. The author plans to add a threshold value to improve the model's accuracy, specificity, and sensitivity.

X Yu and W Pang suggested a pre-trained deep fusion learning model to diagnose benign and normal tumors [33]. The author proposed two deep fusion learning models: model 1 and model 2. The model extracted the ROI from the database. The models have two steps. First, ROI patches were modeled to diagnose normal and tumor, and model 2 integrated the feature using 1*1 convolution. Model 1 has 0.89 accuracy, 0.91 recall, and 0.80 precision. Model 2 gave the tumor 0.87 accuracy, 0.95 recall, and 0.75 precision.

Globally, one woman gets breast cancer every 20 seconds and dies every 5 minutes. Over “170,000” new cancer cases are expected in Pakistan this year. Before it spreads and kills, breast cancer must be stopped. ML is crucial to medical image categorization. In recent years, ML approaches have been developed for manual and automatic disease identification. ML algorithms are advancing, including deep learning. Deep learning (DL) is a smarter, more advanced machine learning field. DL approaches significantly impact medical image classification models [2].

P. Danaee detected breast cancer using deep learning. P. Danaee utilized a stacked denoising auto-encoder to extract gene features and test a supervised classification model to detect cancer with the new features [34]. These two qualities are ideal for early breast cancer screening, according to the author. The author plans to diagnose more breast cancer types using a larger dataset.

D. Selvathi introduced a sparse auto-encoder (SAE)-based breast cancer detection system that is error-free and quick compared to previous methods that learn feature representations from image datasets and classifiers [35]. The SAE performs in Random Forest Classifier, Support Vector Classifier (SVC), and K Nearest Neighbour. Random Forest performed best in this research. After pre-processing on SAE unsupervised learning, the Random Forest model gives 98.9% accuracy on publicly accessible MIAS dataset mammograms.

In [36], Y.J. Tan suggested a convolutional neural network model for early breast cancer diagnosis using pictures. Normal, benign, and malignant tumors exist. Classifying mammograms: MCCNN and BCDCNN improved image classification. Up to 87% system accuracy. 322 mammograms are in the dataset. CNN has the highest accuracy and the fastest diagnosis.

In this reference, Omondiagbe examines support vector classifiers, artificial neural networks, and naïve Bayes approaches on the WCBC database [37]. The author incorporates all machine learning approaches with an early feature selection strategy and examines their performance. The proposed model reduced dataset dimensionality using linear discriminant analysis and a hybrid technique. The model's accuracy is 98.42. The author plans to analyze more breast cancer detection machine learning algorithms and construct a model that predicts other breast cancer-related disorders.

S. Karthik proposed computer-aided detection for early breast cancer analysis utilizing a deep learning neural network with many layers on a support vector machine classifier for improved accuracy [16]. This study uses Wisconsin breast cancer detection. The dataset analysis yielded 98.62% accuracy, better than the other state-of-the-art systems's designs and tests. The author wants to work on particle swarm optimization to save time and enhance accuracy.

A Breast Ultrasound (ABUS) by Y. Wang uses 3D CNN for computer-aided cancer screening [13]. The author claimed to have pioneered 3D CNN deep learning. ABUS provided a 3D representation of the breast and an independent image that a radiologist could understand. The author proposed using 3D CNN for cancer diagnosis in automated breast ultrasound. The author detected cancer with excellent sensitivity and low false positives using a multilayer feature. The voxel-level adaptive threshold distinguishes benign and malignant tumors. The author tested this method on 900 volumes of 745 malignant and 144 healthy women. This experiment yields 95% sensitivity and 0.84 false positives. Breast cancer detection using ABUS is sensitive and has few false positives.

This research employed the DDSM and CBIS-DDSM datasets and tested multiple deep learning models. Deep convolutional neural networks (CNNs) are investigated for breast cancer CAD. CNNs are built and tested on two mammographic datasets, with ROIs showing harmless or worrisome mass sores [12]. The exhibition assessment of each inspected network is done in two ways: with pre-prepared loads and arbitrarily. Broad test findings illustrate the maximum exhibition achieved by adjusting a pre-prepared organization versus preparing without preparation. This research's best models were Alex Net and Resnet 50.

3. Proposed Methods

The process consists of four steps. Grab a dataset first from the UCI Repository. Second Pre-Processing a dataset and extracting its features Third Data splitting, training, and testing come last. Data classification and evaluation using deep learning and machine learning algorithms. Model of deep learning CNN and an analysis of a few logistic regression, SVM, and KNN classifiers for machine learning. The UCI repository's WCBD dataset is used to test machine learning algorithms. In cross-validation tests, the logistic regression classifier model yields the best accuracy, nearly 98.25 percent. Our deep learning CNN model is trained using the Keras technique, with 70% of the data used for training and 30% for testing. The accuracy of the

neural network model l is 97.66%. Model 2 fig 4-3 displays the process for the other two datasets. When compared to the inceptionV3 model on the DDSM dataset, the sequential model's accuracy on the BreakHis dataset was superior. The percentages for testing and training are 20% and 80%, respectively.

The proposed model 1, which we applied to WCBD after acquiring the dataset, is depicted in figure 1 below

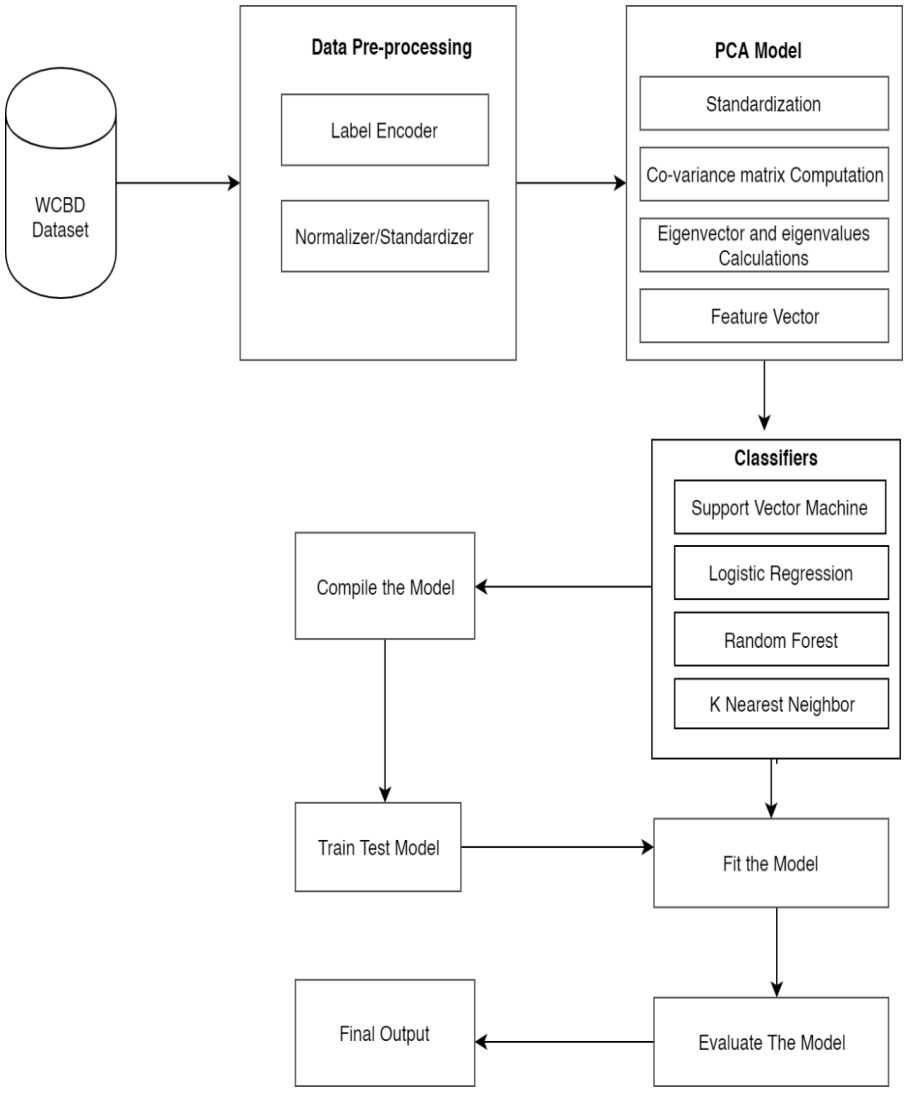

Figure1: Methodology on WCBD Dataset with the PCA (model1a)

In Figure 2, an approach to describing WCBD without the use of a PCA model is demonstrated.

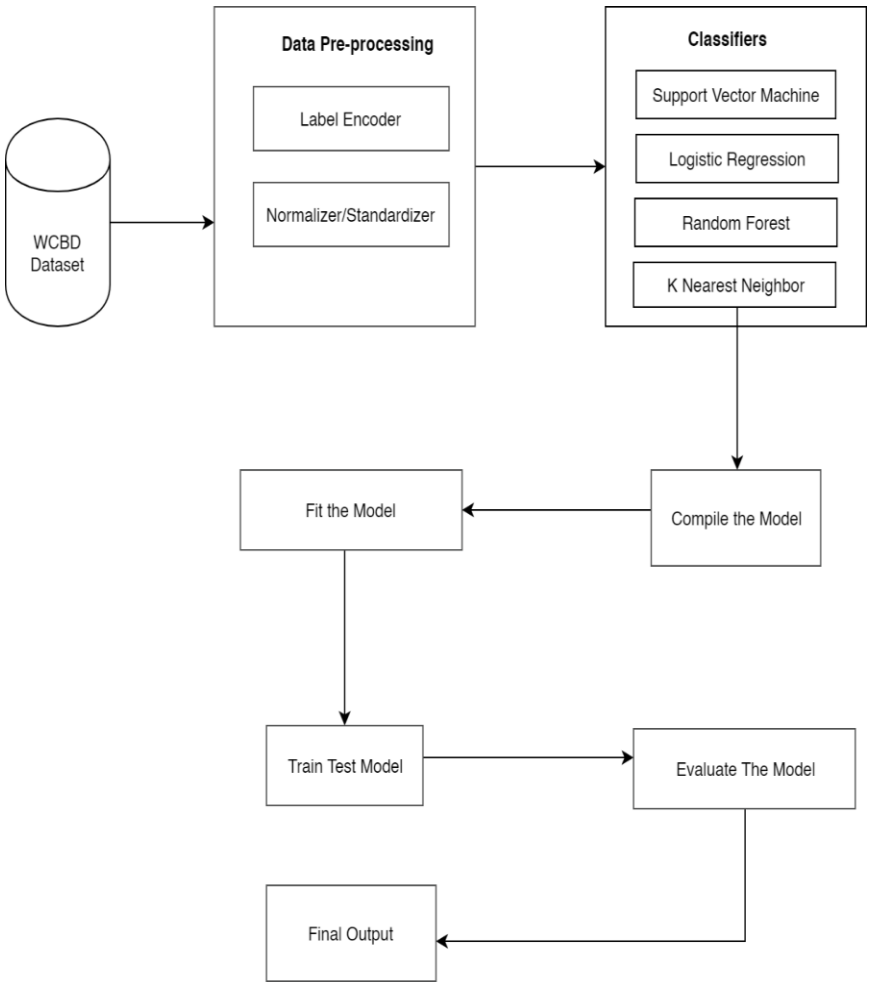

Figure 2: Methodology on WCBD Dataset Without the PCA (model1b)

The methodology of our proposed model2 is displayed in figure 2. We have performed pre-processing on the DDSM and Breakhis datasets, including reshaping and resizing the images. We then applied a CNN model to the DDSM dataset and an InceptionV3 model to the Breakhis dataset. When working with the inceptionV3 and CNN model, it's important to experiment with various layers such as convolution, pooling, and activation functions like Relu. Additionally, the fully connected layer with Sigmoid and SoftMax can be utilized to classify tumors as either benign or malignant.

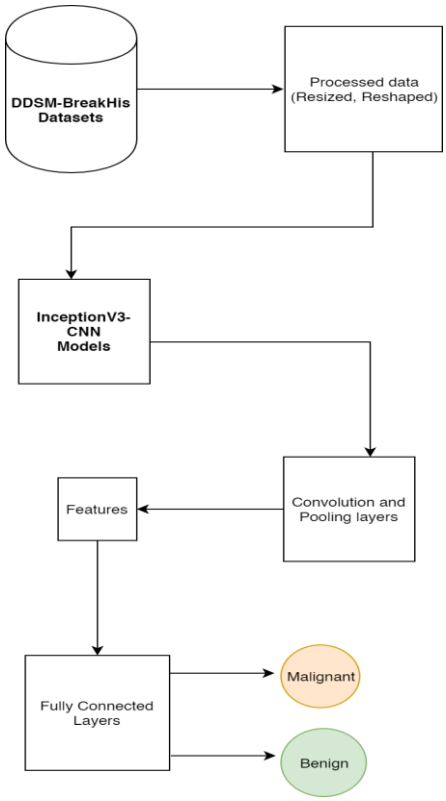

Figure 3: Methodology on DDSM and Breakhis Dataset (model2)

3.1. Datasets

3.1.1. The Wisconsin breast cancer database

The dataset covers 699 instances and 10 attributes. The dataset has a missing value which would be dropout Samples attained occasionally as Dr. Walberg stat in his clinical cases. The dataset replicates this consecutive group of the data. This group information seems immediately below, having been eliminated from the data itself. Benign is a non-cancerous tumour, whereas malignant is a cancerous tumour. We can write benign as 0 and malignant as 1.

3.1.2. DDSM Mammography

The DDSM dataset comprises images, which were pre-processed and subsequently resized to a dimension of 299*299 pixels. 86% of the "55890" training images in this dataset are negative examples, while the remaining 14% are positive examples.

- Pre-processing

There are two categories of images in the DDSM dataset: positive and negative. Positive images are included in the CBIS-DDSM dataset, while negative images are included in the DDSM dataset. During data pre-processing, images are resized to dimensions of 299 by 299 pixels. The negative image was initially 598 by 598 pixels and was subsequently resized to 299 by 299 pixels [11]. In order to extract the region of interest (ROI) from the positive images in the DDSM dataset, a mask with spacing is applied to specify the location of the image. Each image of the region of interest was arbitrarily cropped three times into 598*598 dimensions, with pre-processing including rotation, reversal, and resizing to 299*299.

The resized images are labeled into two binary and multi labels as:

- Binary class - 1 for malignant and 0 for benign

- Multi-Label - 0 for negative, 1 for benign calcification, 3 for malignant calcification, 2 for benign mass, and 4 for malignant mass

3.1.3. Breast Cancer Histopathological Dataset (BreakHis)

There are a total of 7909 images of benign and malignant breast cancer tumors in this dataset. A total of 82 patients were utilized to obtain these microscopic images at various magnification factors. Out of a total of 7909 images, 2480 are benign and 5429 are malignant samples. Each image is 740 by 460 pixels in dimension, in PNG format, and contains an 8-bit depth in the RGB channel [4]. The development of this database was facilitated by the Brazilian organization Pathological Anatomy and Cytopathology. The proprietors of a dataset are certain that this dataset will serve as the optimal resource for their investigations and prove beneficial in the detection of breast cancer.

3.2. Data Preprocessing

3.2.1. Categorical Variable Conversion

Attributing categorical information denotes discrete values that are members of a particular finite set of classes or groups. These are often referred to as "groups" within the context of expected attributes or variables generated by the model. These unique values are either textual or numeric. Definite data can be broadly classified into two classes: ordinal and nominal. The dataset combines categorical and numerical mechanisms. Thus, two distinct sorts of perceptions exist regarding breast cancer. M represents malignancy, while B represents benignly. Each classifier performs admirably with numerical data. Consequently, in order to convert non-numerical data to a numerical value, a "label encoder" was required.

3.2.2. Feature Scaling

When operating with a learning model, scaling the options to a value of variability that is close to zero is dynamic. This may be achieved by ensuring that the variation of the options remains consistent. If the variance of a single feature is orders of magnitude greater than the variance of other options, that feature could potentially result in distinct options within the dataset. We do not want this to happen with our model. The aim at this stage is to attain a mathematical expression that has a mean of zero and a variance of one unit. Although numerous practices exist in this regard, standardization and normalization are the most recent. By substituting their Z scores for the values, standardization occurs.

3.2.3. The Principal Component Analysis (PCA)

The PCA is a technique of correlational analysis that studies the full variance within the knowledge, that is the mutual correlational analysis, and converts the first variables into a minor set of linear mixtures. The diagonal of the matrix covers unions and therefore the full variance is transported into the tissue matrix. The term issue matrix is that the matrix that protects the factor loadings of all the variables on all the factors removed. The term, “factor loadings” is the unassertive correlation between the factors and the variables. The PCA could be a method used for the documentation of a smaller range of distinct variables stated as principal elements from a superior set of information. The technique is wide famine to emphasize variation and detentions well-made patterns in an exceptionally knowledge set. The principal element analysis is recommended once the researcher’s primary anxiety is to work out the minimum range of things that may account for the extreme variance within the knowledge in use in the detailed statistical process, like in city studies. The eigenvalues talk over with the whole variance enlightened by every issue. The quality deviation measures the variability of information. The job of principal part analysis is to advert the patterns within the data and to straighten the info by grace.

PC1 and PC2 represent the PCA model composition in Figure 4. Red spots indicate benign tumors, while green dots indicate malignant tumors. Based on the data presented in Figure 4, it can be observed that the majority of the cases extant at a given point (2.5 0.0) on PC1 and PC2 are benign in nature.

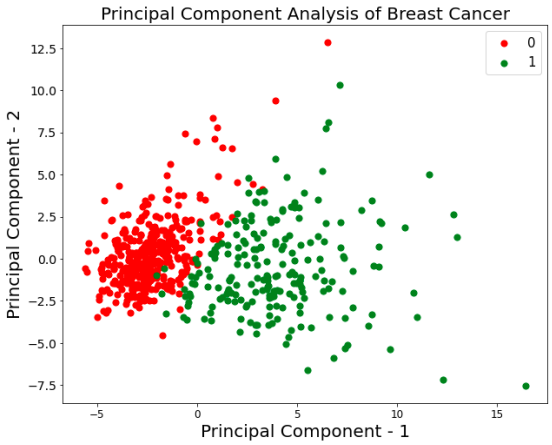

Figure 4: PCA Model Output on WCBD

4. Results and Evaluations

This study evaluates classification issues and classifier performance matrices. The binary variable 1(Malignant) indicates a positive breast cancer diagnosis. A negative instance (0) indicates no breast cancer. In this chapter, we compare machine learning classifier and deep learning model outcomes.

4.1. Models Performance

4.1.1. K-Nearest Neighbors (KNN)

The training set comprises 30% of the model and the evaluation set comprises the remaining 70%, for a total accuracy of 94.24%. However, by implementing Principal Component Analysis (PCA), we were able to improve both accuracy and recall. We attained a 98% recall rate, 99% precision, and 98% accuracy rate. The model obtained an accuracy of 98%, precision of 97%, and recall of 98% for both benign and malignant tumors. The confusion matrix after straightforward KNN and PCA application are compared below. 98% is the aggregate performance of the KNN classifier when applied to the PCA model.

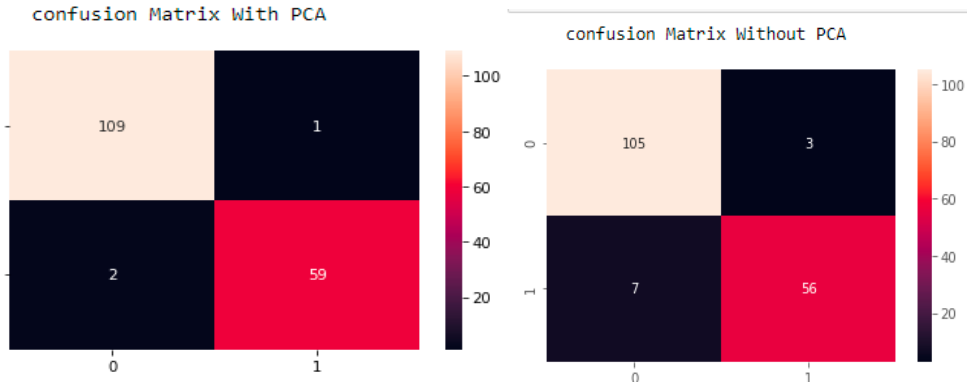

Figure 5: Performance Comparison of the KNN

4.1.2. Logistic Regression (LR)

The Logistic Regression (LR) classifier demonstrates an accuracy rate of 99.4%. Accuracy with the PCA model is 96.49%, whereas accuracy without the PCA model is 96.49%. Figure 5-2a illustrates that the implementation of Principal Component Analysis (PCA) resulted in enhanced precision, recall, and the f1 score. A 98% precision, a 100% recall, and a 99% F1 score. The PCA model obtained a precision of 100%, recall of 97%, and f1 score of 98% for both benign and malignant tumors. 96% is the accuracy of LR in the absence of the PCA model.

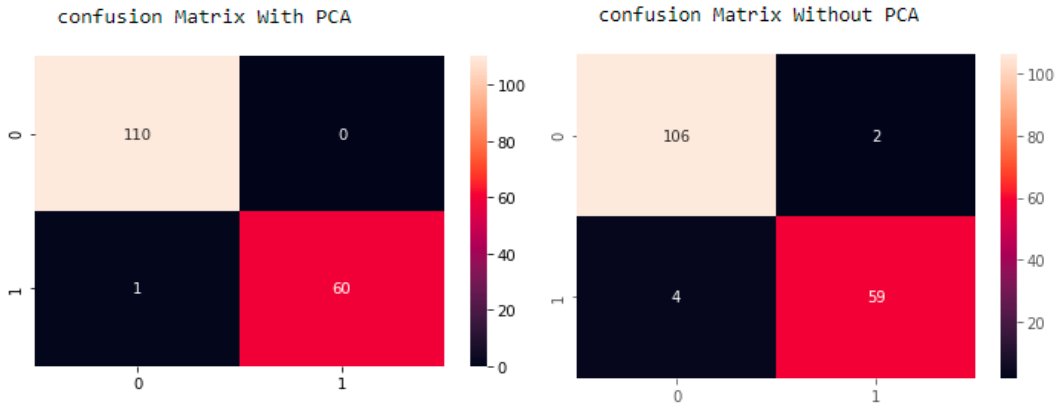

Figure 6: The Performance Comparison of Logistic Regression

4.1.3. Support Vector Machine (SVM)

The Support Vector Machine (SVM) model demonstrates a remarkable accuracy of 99%. The accuracy achieved using the PCA model is 98%, whereas its absence results in a lower accuracy of 98%. Upon implementing Principal Component Analysis (PCA), a marginal increase in the accuracy of the model was observed. The PCA model achieves an accuracy of 99%, precision of 99%, recall of 100%, and f1 score of 100% for benign tumors. For malignant tumors, the model achieves precision of 100%, recall of 98%, and f1 score of 99%.

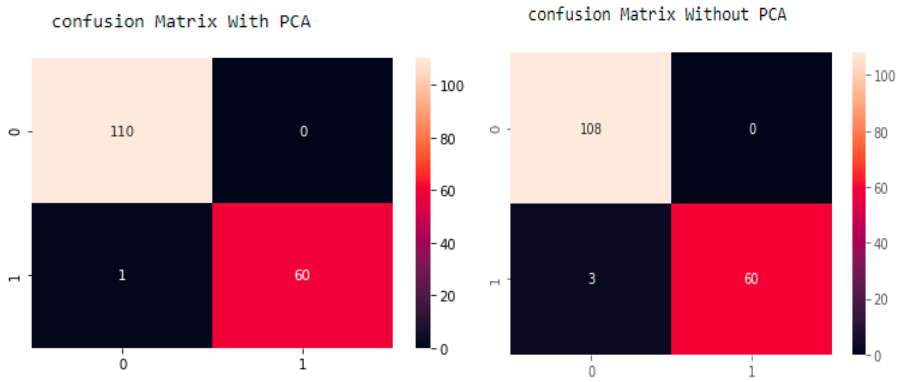

Figure 7: Performance Comparison of SVM

4.1.4. Random Forest (RF)

The Random Forest (RF) model demonstrates a 98% accuracy rate. Achieving an accuracy of 96% without the utilization of the PCA model is possible. We observed that by implementing Principal Component Analysis (PCA), our accuracy, recall, and f1 score all improved. The PCA model obtained an

accuracy of 98.2%, precision of 98%, recall of 99%, and f1 score of 100% for benign tumors, and precision of 98%, recall of 97%, and f1 score of 97% for malignant tumors.

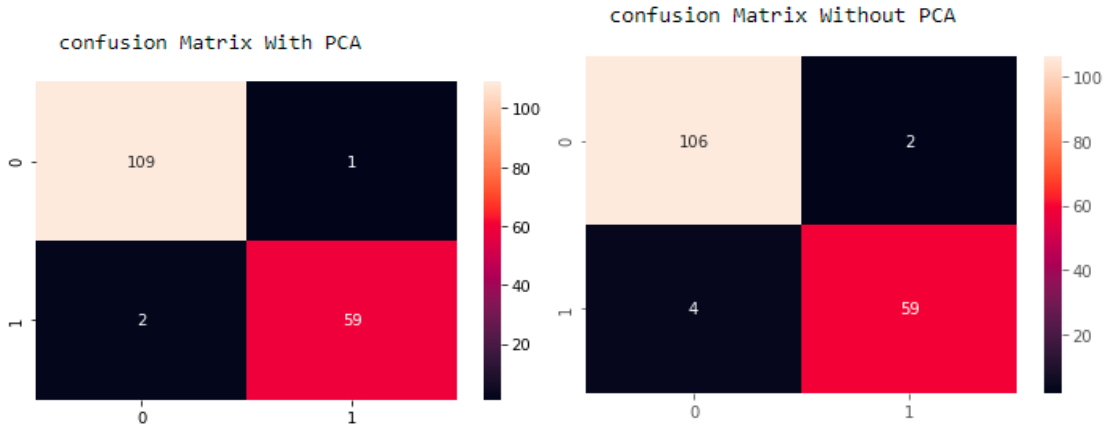

Figure 8: Performance Comparison of Random Forest

4.2. Comparison And Analysis Between the Algorithms

In this research paper, four ML algorithms are examined in depth. In this paper, we compare and contrast the fundamental characteristics and features of four machine learning algorithms. In our experiment, the efficacy of Logistic Regression is comparable to that of other models; both training and prediction are significantly improved by approximately 99 percent.

Table 1 presents comparisons of four machine learning models. Based on the data presented in the table, it can be concluded that the logistic regression model exhibits superior performance compared to the others, as it enables rapid and accurate predictions and has a commendable training speed.

Table 1: Comparison and analysis of ML Algorithms

	Accuracy Prediction	**Training Speed**	**Prediction Speed**	**Performance on a small observation**
Logistic Regression	Fast	Fast	Yes	Yes
Random Forest	High	Slow	Moderate	Yes
KN Neighbors	Low	No Training	Slow	Yes
SVM	High	Slow	Fast	Yes

4.3. Graphical comparison among the algorithms

In this study, we compared four distinct types of machine learning algorithms for accuracy with and without the PCA model. With the PCA model, logistic regression and SVM provided better accuracy than 98%, but without the PCA model, logistic regression and support vector machine produced 99% accuracy, both being the most accurate. The y-axis represents the accuracy rate, while the x-axis represents the algorithm names.

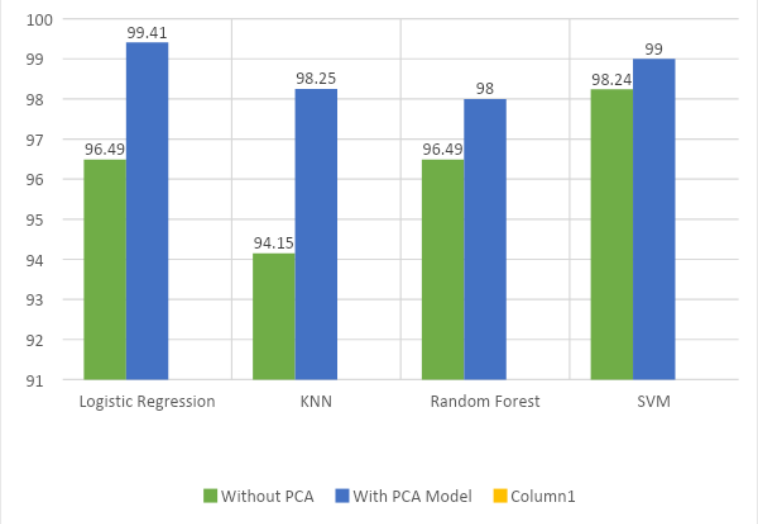

Figure 9: Graphical Representation of B and M Using PCA

Figure 9 illustrates graphical representations of model accuracy. The classification of benign and malignant tumors using various machine learning techniques is demonstrated. The x-axis displays algorithm names, while the y-axis displays accuracy rates. Logistic regression has the highest accuracy of the three classifiers when utilizing the PCA model.

4.4. Performance of Deep Learning Model on WBCD Dataset

The deep learning model performs well on Keras at 96.24%. We employ seven input layers with the activation function Rectified linear unit (Relu), and for output, we attempt two functions. In our scenario, SoftMax outperforms Sigmoid because it excels at binary categorization. We divide the dataset into two portions with varying weightages: 80% for training and 20% for testing. We run these layers for 50 epochs with a "128" batch size, and the model achieves 96.24% accuracy.

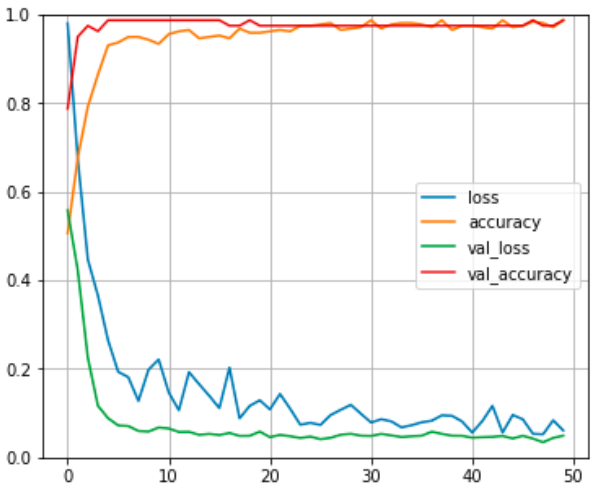

Figure 10: Accuracy Graph of DL Model on WCBD

Figure 10 shows that a deep learning model achieves 96% accuracy on the WCBD dataset. The value loss ranges from 0.2 to 0.3, the loss from 0.0 to 0.6, and the validation accuracy from 0.65 to 0.99, as indicated in the Accuracy Graph.

4.5. DDSM Dataset Performance

Following data pre-processing, we use the Inceptionv3 model to train our data using several input and output layers. In order to enhance the accuracy of a model, the training step incorporates the inclusion of hidden layers. 80% of the data is allocated for training, while the remaining 20% is reserved for testing. The activation function employed for the output is the sigmoid layer. Achieving an accuracy of 95.73% and a f1 score of 95.67% was seen across 36 epochs and 128 batch sizes. However, prior to our pre-processing and model training, the accuracy of Inceptionv3 was 92.22%. This accuracy subsequently improved to about 3%, which is considered satisfactory for predicting breast cancer tumors. Below are the classification reports of our model, as well as the results obtained from our training and feature extraction.

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	9719
1	0.85	0.82	0.83	1458
accuracy			0.96	11177
macro avg	0.91	0.90	0.90	11177
weighted avg	0.96	0.96	0.96	11177

Figure 11: Classification Report After the Experiment

Figure 11 demonstrates a model accuracy of 96%, surpassing the accuracy of the initial InceptionV3 model, which was approximately 92%. Based on the findings presented in figure 11, it can be inferred that our trained model demonstrated enhanced precision in predicting and diagnosing breast cancer tumors as either benign or malignant.

Classification Report:

	precision	recall	f1-score	support
0	0.98	0.93	0.95	14579
1	0.65	0.89	0.75	2187
accuracy			0.92	16766
macro avg	0.81	0.91	0.85	16766
ghted avg	0.94	0.92	0.93	16766

Figure 12: Classification Report before the Experiment

Figure 12 displays the pre-training performance of the InceptionV3 model on the DDSM dataset. The training of InceptionV3 by the other author yields an average accuracy of 92%.

4.5.1. InceptionV3 Confusion Matrix on the DDSM Dataset

The Inception V3 model's confusion matrix on the DDSM dataset, as shown in figure 13, indicates that the model achieves an accuracy of 96% on this dataset. There were 208 false positives and 269 false negatives out of 11777 scan pictures, which are represented by the diagonal value.

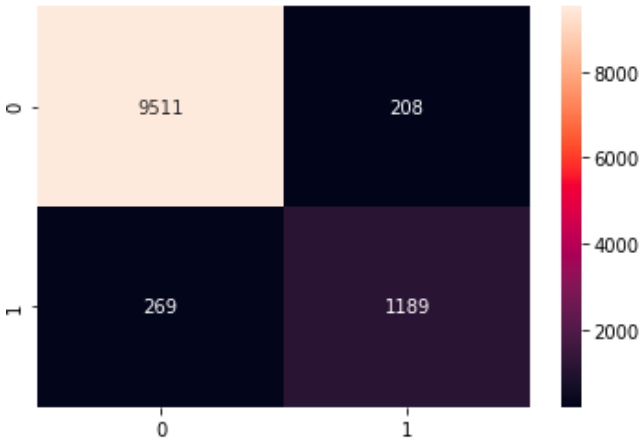

Figure 13: Confusion Matrix on DDSM Dataset

4.5.2. Graphical representation of Accuracy on DDSM dataset

Accuracy, validation accuracy, loss, and validation loss are graphically depicted in figure 14. With a validation loss of more than 0.2, a model's accuracy stays over 96%.

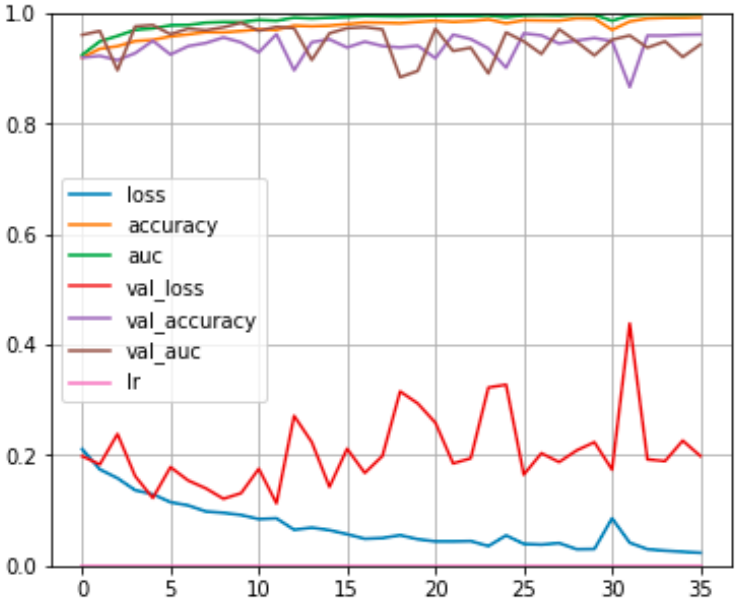

Figure 14: Graphical representation of Accuracy on DDSM dataset

4.6. Comparison of different Deep Learning models on DDSM dataset

Here in Table 5, we take a look at the DDSM Dataset and compare five distinct deep learning models. All of the models' average f1 and precision-recall scores are displayed. Our trained model, inceptionv3, outperformed the other four models with an accuracy of over 96%. Our model for the DDSM dataset was a pre-trained InceptionV3. Following picture pre-processing and segmentation, the model is executed on hundreds of photos. Author Li Shen discusses Resnet50 and VGG-16 in this study [10], although the inceptionV3 model outperformed both in terms of accuracy.

Table 5: Deep Learning Models Performance ON DDSM

Model Name	Accuracy	Precision	Recall	F1score
Inceptionv3	**96**%	**95**%	**95**%	**96**%
Resnet50	89%	89%	88%	88%
Densenet121	91%	90%	91%	90%
Mobile Net	90%	89%	90%	88%
VGG-16	84%	85%	83%	84%

4.7. Breakhis Dataset Performance

Once the data has been pre-processed, it is trained using a deep learning model with several input and output layers. To further enhance a model's accuracy, some hidden layers are also incorporated during the training phase. For training, we use 80% of the data, and for testing, we use 20%. As an activation function for output, the SoftMax layer is utilized. We get a 98.8% accuracy and a 99% F1 score over 5 epochs with 128 batch sizes, which is sufficient for predicting tumors associated with breast cancer.

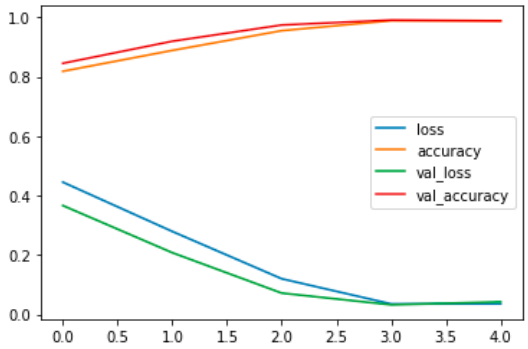

Figure 15: Accuracy Graph of BreakHis Dataset

In Figure 15, we can observe the accuracy curve of the BreakHis dataset on CNN. Both the validation accuracy and the accuracy on benign and malignant tumors reach 99%. From 0.4 to 0.01, loss and validation loss are decreasing at a steady rate.

Classification Report:

	precision	recall	f1-score	support
0	0.98	0.99	0.98	2019
1	0.99	0.99	0.99	4308
accuracy			0.99	6327
macro avg	0.98	0.99	0.99	6327
weighted avg	0.99	0.99	0.99	6327

Figure 16: Classification Report of BreakHis dataset

Figure 16 displays classification reports on the BreakHis dataset, which inform us that, out of a total of 6327 pictures, 2019 are predicted to be benign with a 98% precision and 4308 as malignant with a 99% precision.

4.7.1. Confusion Matrix of CNN model on BreakHis Dataset

Figure 17 displays the confusion matrix of the BreakHis dataset. It is evident that the diagonal location of the matrix exhibits a relatively low number of images, indicating a high level of accuracy in our predictions. Out of the total of 7909 photos, a mere 25 instances are classified as false positives, indicating that the model incorrectly predicts the presence of breast cancer in 25 patients who do not actually have breast cancer. Conversely, 51 instances are classified as false negatives, indicating that the model incorrectly predicts the absence of breast cancer in 51 patients who actually have breast cancer.

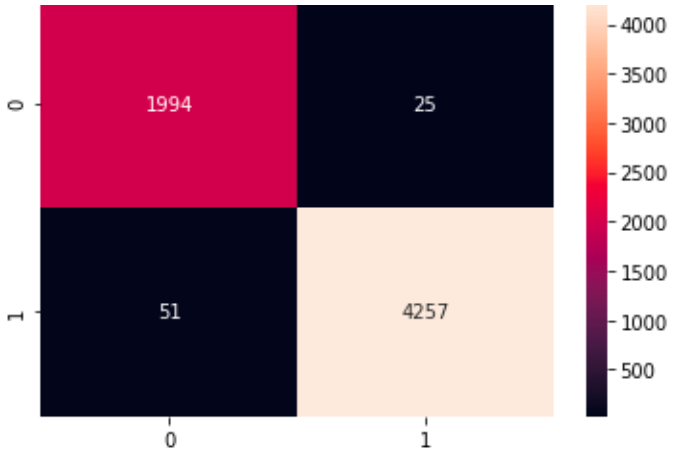

Figure 17: Confusion Matrix of CNN model on BreakHis Dataset

5. Conclusion

In conclusion, our research highlights the significant significance of prompt and precise breast cancer detection, particularly in poor countries where women have obstacles in accessing healthcare services. By utilizing advanced deep learning models like InceptionV3 and Sequential, we were able to attain impressive classification accuracies on several datasets. Specifically, we achieved a classification accuracy of 96.24% on WCBD and 98.8% on BreakHis. The investigation conducted on supervised machine learning techniques serves to underscore the effectiveness of deep learning methodologies, wherein Logistic Regression demonstrates greater performance. This research makes a valuable contribution to the advancement of healthcare outcomes on a global scale by highlighting the importance of strong diagnostic tools and the possibility of deep learning in enhancing breast cancer detection.

Future work in breast cancer detection might involve refining deep learning algorithms for use with parallel processing systems and publicly available medical picture datasets in order to increase accuracy, and the integration of physician verification to better safeguard patient care.

References

[1] Mehreen Tariq, Sajid Iqbal, Hareem Ayesha, Ishaq Abbas, Khawaja Tehseen Ahmad, and Muhammad Farooq Khan Niazi. "Medical image based breast cancer diagnosis: State of the art and future directions." *Expert Systems with Applications* 167 (2021): 114095.

[2] Debelee, Taye Girma, Friedhelm Schwenker, Achim Ibenthal, and Dereje Yohannes. "Survey of deep learning in breast cancer image analysis." *Evolving Systems* 11, no. 1 (2020): 143-163.

[3] Ragab, Dina A., Maha Sharkas, Stephen Marshall, and Jinchang Ren. "Breast cancer detection using deep convolutional neural networks and support vector machines." *PeerJ* 7 (2019): e6201.

[4] Nawaz, Majid, Adel A. Sewissy, and Taysir Hassan A. Soliman. "Multi-class breast cancer classification using deep learning convolutional neural network." *Int. J. Adv. Comput. Sci. Appl* 9, no. 6 (2018): 316-332.

[5] Fathy, Wael E., and Amr S. Ghoneim. "A deep learning approach for breast cancer mass detection." *International Journal of Advanced Computer Science and Applications* 10, no. 1 (2019).

[6] Khan, A. A., and A. S. Arora. "Breast cancer detection through gabor filter based texture features using thermograms images." In *2018 First international conference on secure cyber computing and communication (ICSCCC)*, pp. 412-417. IEEE, 2018.

[7] Cheng, Heng-Da, Xiaopeng Cai, Xiaowei Chen, Liming Hu, and Xueling Lou. "Computer-aided detection and classification of microcalcifications in mammograms: a survey." *Pattern recognition* 36, no. 12 (2003): 2967-2991.

[8] Abdel-Nasser, Mohamed, Antonio Moreno, and Domenec Puig. "Breast cancer detection in thermal infrared images using representation learning and texture analysis methods." *Electronics* 8, no. 1 (2019): 100.

[9] Paquin, Francis, Jonathan Rivnay, Alberto Salleo, Natalie Stingelin, and Carlos Silva. "Multi-phase semicrystalline microstructures drive exciton dissociation in neat plastic semiconductors." *arXiv preprint arXiv:1310.8002* (2013).

[10] Shen, Li. "End-to-end training for whole image breast cancer diagnosis using an all-convolutional design." *arXiv preprint arXiv:1711.05775* (2017).

[11] Lee, Rebecca Sawyer, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy, and Daniel L. Rubin. "A curated mammography data set for use in computer-aided detection and diagnosis research." *Scientific data* 4, no. 1 (2017): 1-9.

[12] Tsochatzidis, Lazaros, Lena Costaridou, and Ioannis Pratikakis. "Deep learning for breast cancer diagnosis from mammograms—a comparative study." *Journal of Imaging* 5, no. 3 (2019): 37.

[13] Wang, Yi, Na Wang, Min Xu, Junxiong Yu, Chenchen Qin, Xiao Luo, Xin Yang, Tianfu Wang, Anhua Li, and Dong Ni. "Deeply-supervised networks with threshold loss for cancer detection in automated breast ultrasound." *IEEE transactions on medical imaging* 39, no. 4 (2019): 866-876.

[14] Sree, Subbhuraam Vinitha, Eddie Yin-Kwee Ng, Rajendra U. Acharya, and Oliver Faust. "Breast imaging: a survey." *World journal of clinical oncology* 2, no. 4 (2011): 171.

[15] Jaafar, Bochra, Hela Mahersia, and Zied Lachiri. "A survey on deep learning techniques used for breast cancer detection." In *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1-6. IEEE, 2020.

[16] Karthik, S., R. Srinivasa Perumal, and P. V. S. S. R. Chandra Mouli. "Breast cancer classification using deep neural networks." *Knowledge Computing and Its Applications: Knowledge Manipulation and Processing Techniques: Volume 1* (2018): 227-241.

[17] Karthikeyan, B., Sujith Gollamudi, Harsha Vardhan Singamsetty, Pavan Kumar Gade, and Sai Yeshwanth Mekala. "Breast cancer detection using machine learning." *Int. J. Adv. Trends Comput. Sci. Eng* 9, no. 2 (2020): 981-984.

[18] Khuriwal, Naresh, and Nidhi Mishra. "Breast cancer diagnosis using deep learning algorithm." In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 98-103. IEEE, 2018.

[19] Khasana, Uswatun, Riyanto Sigit, and Heny Yuniarti. "Segmentation of breast using ultrasound image for detection breast cancer." In *2020 International Electronics Symposium (IES)*, pp. 584-587. IEEE, 2020.

[20] Chouhan, Naveed, Asifullah Khan, Jehan Zeb Shah, Mazhar Hussnain, and Muhammad Waleed Khan. "Deep convolutional neural network and emotional learning based breast cancer detection using digital mammography." *Computers in Biology and Medicine* 132 (2021): 104318.

[21] Rathi, Megha, and Vikas Pareek. "Hybrid approach to predict breast cancer using machine learning techniques." *International Journal of Computer Science Engineering* 5, no. 3 (2016): 125-136.

[22] Bharat, Anusha, N. Pooja, and R. Anishka Reddy. "Using machine learning algorithms for breast cancer risk prediction and diagnosis." In *2018 3rd International conference on circuits, control, communication and computing (I4C)*, pp. 1-4. IEEE, 2018.

[23] Jannesari, Mahboubeh, Mehdi Habibzadeh, HamidReza Aboulkheyr, Pegah Khosravi, Olivier Elemento, Mehdi Totonchi, and Iman Hajirasouliha. "Breast cancer histopathological image classification: a deep learning approach." In *2018 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pp. 2405-2412. IEEE, 2018.

[24] Yilmaz, Feyza, Onur Kose, and Ahmet Demir. "Comparison of two different deep learning architectures on breast cancer." In *2019 Medical Technologies Congress (TIPTEKNO)*, pp. 1-4. IEEE, 2019.

[25] Goni, Md Omaer Faruq, Fahim Md Sifnatul Hasnain, Md Abu Ismail Siddique, Oishi Jyoti, and Md Habibur Rahaman. "Breast cancer detection using deep neural network." In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pp. 1-5. IEEE, 2020.

[26] Kathale, Poonam, and Snehal Thorat. "Breast cancer detection and classification." In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1-5. IEEE, 2020.

[27] Kumari, Madhu, and Vijendra Singh. "Breast cancer prediction system." *Procedia computer science* 132 (2018): 371-376.

[28] Gerald, S. Z. E. "DETECTION OF BREAST CANCER WITH ELECTRICAL IMPEDANCE MAMMOGRAPHY." (2012).

[29] Varma, P. Satya Shekar, Sushil Kumar, and K. Sri Vasuki Reddy. "Machine learning based breast cancer visualization and classification." In *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)*, pp. 1-6. IEEE, 2021.

[30] Singla, Chaitanya, Pradeepta Kumar Sarangi, Ashok Kumar Sahoo, and Pramod Kumar Singh. "Deep learning enhancement on mammogram images for breast cancer detection." *Materials Today: Proceedings* 49 (2022): 3098-3104.

[31] Wang, Zhiqiong, Mo Li, Huaxia Wang, Hanyu Jiang, Yudong Yao, Hao Zhang, and Junchang Xin. "Breast cancer detection using extreme learning machine based on feature fusion with CNN deep features." *IEEE Access* 7 (2019): 105146-105158.

[32] Khorshid, Shler Farhad, and Adnan Mohsin Abdulazeez. "Breast cancer diagnosis based on k-nearest neighbors: a review." *PalArch's Journal of Archaeology of Egypt/Egyptology* 18, no. 4 (2021): 1927-1951.

[33] Yu, Xiangchun, Wei Pang, Qing Xu, and Miaomiao Liang. "Mammographic image classification with deep fusion learning." *Scientific Reports* 10, no. 1 (2020): 14361.

[34] Danaee, Padideh, Reza Ghaeini, and David A. Hendrix. "A deep learning approach for cancer detection and relevant gene identification." In *Pacific symposium on biocomputing 2017*, pp. 219-229. 2017.

[35] Selvathi, D., and A. AarthyPoornila. "Performance analysis of various classifiers on deep learning network for breast cancer detection." In *2017 International Conference on Signal Processing and Communication (ICSPC)*, pp. 359-363. IEEE, 2017.

[36] Tan, Y. J., K. S. Sim, and Fung Fung Ting. "Breast cancer detection using convolutional neural networks for mammogram imaging system." In *2017 International Conference on Robotics, Automation and Sciences (ICORAS)*, pp. 1-5. IEEE, 2017.

[37] Omondiagbe, David A., Shanmugam Veeramani, and Amandeep S. Sidhu. "Machine learning classification techniques for breast cancer diagnosis." In *IOP conference series: materials science and engineering*, vol. 495, p. 012033. IOP Publishing, 2019.

Machines and Algorithms
http://www.knovell.org/mna

Research Article

Classifiers voting based Decision Support System for Prediction of Kidney Related Chronic Diseases

Mubeen Aslam[1, *], Sajid Iqbal[2] and Ahmad Abdullah[2]

[1]Department of Computer Science and Engineering, University of Engineering and Technology, Lahore, 54000, Pakistan

[2]Department of Computer Science, Bahauddin Zakariya University, Multan, 60000, Pakistan

[*]Corresponding Author: Mubeen Aslam. Email: mubeen.aslam591@gmail.com

Received: 26 April 2023; Revised: 14 June 2023; Accepted: 20 July 2023; Published: 16 August 2023

AID: 002-02-000022

Abstract: Chronic kidney diseases are increasing exponentially due to hypertension, diabetes, anemia and other related factors. Patients with such diseases usually remain unaware of initial symptoms leading to difficulties in diagnosis of the disease. High performance data mining-based diagnosis and prediction techniques could assist the patient in self-analysis and medical practitioners in developing a precise opinion about patient. This research presents a framework for clinical decision support system of chronic kidney disease (CKD) on the basis of knowledge and facts provided by specialists and experts. To diagnose the disease and decide about progression stage of CKD, different classification algorithms are applied and evaluated on the dataset. The proposed methodology increases the accuracy to 91.75 % and reduces the cost of predicting the stages of CKD using LMT algorithms on the dataset.

Keywords: Chronic kidney disease (CKD); Features mining; Classifier fusion; E-Health;

1. Introduction

Chronic Kidney Disease (CKD) is a durable condition where the kidneys cannot work as expected. According to a study, 1.2 million people died worldwide in 2018 [1], and in Western countries, 5-12% of patients have CKD [2]. Due to the high prevalence of CKD, significant treatment expenses, and inconsistent access to treatment, patients and their families face numerous financial and ethical issues. Creatinine is a crucial measurement for diagnosing CKD; it is a chemical waste product created by muscle metabolism. It's normal range is 1.2mg/dl and 1.46mg/dl for women and men correspondingly however a higher amount of creatinine is produced in the later stages of CKD [3]. Chronic diseases are noncommunicable (NCDs), meaning they do not transfer from one person to another, whereas communicable diseases (CDs) can spread from one person to another and replicate quickly. Chronic disease originates from behavioral, biological, social and environmental factors and can lead to death [8,20]. Such disease can be found throughout the world and among all age groups. The human kidney plays a vital role in the body, and diseases related to it are chronic in nature as well. The major function of a kidney is to filter the blood using millions of nephrons to remove the unwanted chemicals and throw them out of the human body. Non-excretion of these unwanted materials leads to chronic disease in the kidney [9].

Chronic Kidney Disease (CKD) [11] is a resilient disease that has five stages, i.e., CKD stage1 to CKD stage5 and could be diagnosed by several parameters such as high blood pressure, diabetes [10] and anemia. Diabetes increases the sugar level of blood that causes injury to the nerves as well as narrows the vessels [14]. Anemia causes high blood pressure, a shortage of red blood cells (RBCs), and low levels of hemoglobin. Red blood cells provide oxygen to body tissues. The provision of lower oxygen can lead to anemia disease. Anemia decreases iron, red blood cells and changes the shape of RBC. Anemia is a hemoglobin combination with having normal range is less than 12 g/dl and 13 g/dl in women and in men respectively [16]. Another estimate [12] describes that more than hundred peoples per million are affected by kidney diseases alone. According to recent statistics (2019) of the National Kidney Foundation (NKF) [13, 26], USA, the mortality rate of CKD is higher than breast cancer or prostate cancer. It is estimated that only in USA, 37 million people and approximately 90% of those who have CKD don't even know about it. Alarming thing is that around 80 million people are at risk for CKD. Recently, it has been seen that people with kidney disease and transplant recipients are at higher risk for developing serious complications from COVID-19.

To determine the warning signs of CKD, feature-based classification can be performed using machine learning methods. Classification can be done using different attributes of available data with data mining tools to diagnose the disease and extract other relevant information. In order to provide better clinical results to practitioners, expert systems are useful as they can perform automated predictions based on patients' available data. It has been proven in many cases that expert systems can perform better than human specialists due to multidimensional data processing [47, 48]. Various data mining techniques are being used to extract knowledge from existing databases and identify comparative data that could be used in decision-making, assessment, and forecasting [17]. Mostly, descriptive and predictive models are used in data mining. Descriptive models categorize patterns to investigate the properties and relations of data, whereas predictive models predict results from various data sources. Both categories of existing data mining models direct towards various tasks such as prediction, classification, association rule mining, clustering, regression, and time-sequence analysis. They further confirm the actual prediction by using classification and clustering. The state of data can make the problem more complicated; for example, the presence of noise, missing labels, dynamic, and large datasets. Issues with datasets decrease their performance when used in machine learning algorithms.

Similar issues are raised when working with medical datasets to extract unknown patterns and identify the extracted pattern. In the routine diagnosis process, common issues in bioinformatics are not handled properly. Practitioners recommend different test procedures to find out deep information about the disease and formulate their diagnosis. If the set of tests is not formed considering different aspects of the disease, it usually complicates the diagnosis process. Even the use of multiple tests may divert from the correct diagnosis procedure [4], increasing the treatment cost and reducing the performance of prediction methods. These problems can be reduced by using machine learning algorithms that may overcome data deficiencies easily [5]. Classification algorithms are popular in healthcare applications and are used to diagnose and predict the disease in earlier stages [6]. Another aspect that has made classification popular among practitioners is the ability of machine learning methods to deal with complex and large datasets.

Currently, healthcare researchers and industries are applying state-of-the-art statistical methods to assist and guide medical practitioners in treating a wide range of diseases. Statistical methods either use handcrafted features from medical data or automatically extract the features and use them in their decision-making [22]. If the features are handcrafted, the quality of the results depends on the quality of the features used, and the standard of features depends on the knowledge and expertise of the algorithm designer.

The significance of this research is public health impact of CKD, the potential of machine learning in healthcare, and the importance of early detection and intervention in improving patient outcomes in real-time environment.

In this study, we aimed to predict the stages of chronic kidney disease by extracting features from a given dataset. Subsequently, we developed a decision support system to assist both patients and doctors. This system provides information that may not be readily accessible to human experts, without any time

loss. Our goal was to enhance algorithm efficiency and accuracy. To achieve this, we employed a variety of algorithms including logistic model tree, functional tree, J48, Naïve Bayes, Support Vector Machine (SVM), and Artificial Neural Networks (ANN). These algorithms offer valuable insights that can aid physicians in predicting chronic kidney disease at early stages. We utilized a multi-class benchmark dataset and patient history to train the classifiers, fine-tuning various parameters in the process.

The rest of the paper is organized into four sections, Section 2 reports the literature review of different diseases, i.e., diabetes, heart and kidney diseases. Section 3 presents classification methods in detail used in the current era. Section 4, describes the proposed methodology and architecture. Section 5, presents results and discussion, finally, Section 6 concludes research along with future directions.

2. Literature Review

Biological data has experienced significant growth due to multiple factors such as advancements in recording mediums, automated data generation methods, the expansion of medical facilities, and the increase in the human population. Despite this growth, the mortality rate due to various diseases is also on the rise. One major reason for this increased mortality rate is the failure to detect diseases at their initial stages. Some diseases are particularly challenging to diagnose early, including chronic kidney disease (CKD), which progresses slowly alike to kidney failure, cancer, heart disease, asthma, and diabetes. In recent years, numerous classification tasks have been undertaken to predict chronic diseases. Classification algorithms play a crucial role in enhancing the accuracy of disease prediction, with research efforts focused on improving diagnostic accuracy based on clinically collected information. The aim is to detect diseases at early stages to facilitate the development of better treatment options. Popular machine learning algorithms employed in AI-based healthcare systems include Support Vector Machine (SVM), Artificial Neural Networks (ANN), k-Nearest Neighbors (KNN), and Random Forest (RF).

Diabetes leads to various health complications such as heart disease, kidney failure, blindness, and stroke. Hamedan et al. [21] addresses chronic kidney disease (CKD), emphasizing its subtle symptoms and significant healthcare costs. Three phases were undertaken: identifying variables for the Fuzzy Expert System (FES), developing the FES prototype, and evaluating its robustness with noisy data. Initially, 42 parameters were identified from literature and nephrologist consultation, with seven excluded. Key diagnostic parameters included age, blood pressure, proteinuria, and various biochemical markers. The FES achieved high accuracy 92.12% with demonstrated robustness against noisy data. Additionally, Yadollahpour et al. [15] presents an Expert Medical Decision Support System (MDSS) utilizing an Adaptive Neuro-Fuzzy Inference System (ANFIS) to predict the progression of chronic kidney disease (CKD). CKD's covert early stages often delay diagnosis, emphasizing the need for accurate prediction tools to prevent renal damage. The MDSS, based on 10-year clinical records of newly diagnosed CKD patients, predicts Glomerular Filtration Rate (GFR) values, crucial for identifying renal failure. ANFIS, chosen over other models due to its superior accuracy, accurately forecasts GFR variations over 6, 12, and 18-month intervals. The MDSS's performance, evaluated against real patient data, demonstrates high accuracy and efficiency, crucial for improving CKD management and patient outcomes. The user-friendly interface empowers medical professionals with predictive capabilities, enhancing decision-making and patient care. The study underscores the significance of early CKD diagnosis and the potential of ANFIS-based MDSS in improving healthcare outcomes.

Norouzi et al. [28] introduced a medical decision monitoring system for diagnosing kidney failure progression over time. They utilized the Adaptive Neuro-Fuzzy Inference System (ANFIS) algorithm to detect kidney failure progression [16], thereby reducing time costs. The dataset, collected from the hospital, consists of 10 attributes used to predict kidney failure. These attributes include age, weight, Glomerular Filtration Rate (GFR), underlying diseases, calcium, creatinine, sex, diastolic blood pressure, phosphorus, and uric acid. The dataset comprises 465 instances, with 277 being male. The reported accuracy of ANFIS is 95%. Charleonnan et al. [9] discussed the application of machine learning techniques in detecting chronic kidney disease (CKD) to aid clinical practices. They employed algorithms such as Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and Decision Tree (DT) to determine the presence of CKD in patients.

The dataset was divided into training and testing sets with a 70:30 ratio. Through five-fold cross-validation, they reported the average accuracy, with SVM achieving the highest accuracy of 98.3%.

To improve the performance of medical systems and reduce mortality rates, Polat et al. [7] proposed an SVM-based method. They utilized online open-source datasets and collected 400 instances with 24 attributes from the UCI machine repository. Feature selection techniques were employed to reduce data dimensionality. The classifier produced binary results predicting the presence or absence of CKD. The authors also utilized SVM with best-first search and achieved 98.5% accuracy using 10-fold cross-validation.

Ahmad et al. [27] addressed various diseases, their symptoms, and major risk factors affecting kidney patients. The study aims to develop a decision support system for doctors to diagnose kidney disease patients. They employed different data mining techniques such as Naïve Bayes, KNN, and Logistic Regression (LR) to predict kidney diseases, demonstrating better performance. Their methodology was based on classification modeling and the development of an expert system. Steps included data collection, preprocessing, and classification, resulting in a reported accuracy of 98.34%. Rodrigues et al. [10] studied the consequences of dialysis treatment, specifically Continuous Ambulatory Peritoneal Dialysis (CAP), for kidney patients. They compiled a dataset containing records of 850 patients over an 8-year period. Naïve Bayes, KNN, Logistic Regression (LR), Multilayer Perceptrons (MLP), and Random Tree (RT) classifier algorithms were employed. K-NN was identified as the best performer among the classifiers, achieving 99.65% accuracy.

Developing specific datasets is a time-consuming and laborious task. While some researchers create their own datasets, many utilize existing datasets provided through open-source licenses. In [49], Subasi et al. conducted binary classification of CKD using an open-source dataset extracted from the UCI online repository. The dataset comprised 400 instances with 24 attributes. Random Forest (RF), ANN, K-NN, and SVM algorithms were employed, with RF achieving the highest performance accuracy at 99.87%.

Another binary classification work related to CKD detection is presented in [45]. Similar to Thiyagaraj et al., the authors obtained their dataset from the UCI online machine learning repository. The dataset includes instances of diabetes, high blood pressure, cardiovascular disease, and family history of kidney failure, categorized into positive and negative features. Preprocessing and clustering techniques were applied for detection and prediction of CKD.

Detection and diagnosis of CKD patients were performed by Mohamed Elhosney et al. in [47]. They collected data from the UCI online repository and applied two classification algorithms: ant-colony-based optimization (D-ACO) and particle swarm optimization (PSO) using 10-fold cross-validation. D-ACO outperformed other methods, achieving 87.5% accuracy. They further analyzed their results based on various aspects such as precision, F-Score, kappa value, and sensitivity, considering the given datasets.

2.1. Comparison in state of the art

To compare our proposed work with existing research, we have defined a set of parameters including dataset size, type of kidney disease, machine learning algorithm used, achieved accuracy, and classification type. Table 1 presents the comparative analysis of existing studies with our proposed work.

Table 1: Comparison of the different dataset with the proposed dataset

Work	Dataset Specification	Kidney Disease Type	ML Algorithms used	Results (accuracy) (Best classifier)	Classification Type
[35]	Instances: 584 Attributes: 6	4-staged kidney disease	Naïve Bayes, Support Vector Machine	SVM with 76.32%.	Multi-class with 5 classes.
[25]	Instances: 584 Attributes: 6	4-staged kidney disease	ANN, Support Vector Machine	ANN with 87.70% accuracy	Multi-class with 5 classes.

[28]	Instances: 465 Attributes: 10	Kidney Failure progression	Adaptive Neuro-Fuzzy Inference system (ANFIS)	ANFIS with 95% accuracy	Binary classification
[9]	Instances: 400 Attributes: 24	Presence or absence of CKD	SVM, Decision Tree, K-NN and Logistic regression	SVM gives higher performance with 98.3%	Binary classification
[7]	Instances: 400 Attributes: 24	Presence or absence of CKD	SVM with wrapped and filtered evaluator in best first search and greedy step wise	SVM filtered best first search gives 98.5% accuracy.	Binary classification
[10]	Instances: 850 Attributes: 8	Kidney dialysis	LR, MLP, Random tree, Naïve Bayes, K-NN	K-NN attains 99.65% accuracy.	Binary Classification
[46]	Instances: 400 Attributes: 25	Presence or absence of CKD	D-ACO and PSO	D-ACO with 87.5% accuracy	Binary Classification
Proposed dataset	Instances: 800 Attributes: 25	5-stages CKD	ANN, SVM, Naïve Bayes, J48, LMT, FT	LMT has a greater accuracy of 91.375% than other classifiers.	Multi-class with 6 classes

*Stage 1 CKD: eGFR 90 or Greater, Stage 2 CKD: eGFR Between 60 and 89, Stage 3 CKD: eGFR Between 30 and 59, Stage 4 CKD: eGFR Between 15 and 29, Stage 5 CKD: eGFR Less than 15

Table 1 illustrates that the proposed methodology outperforms other works in several aspects. For instance, Vijayarani et al. [35] diagnosed different stages of kidney diseases using 584 instances with 6 parameters and employed Naïve Bayes and SVM classifier algorithms. They achieved a greater accuracy of 76.32% in multi-class classification among 5 classes using SVM. Similarly, Vijayarani et al. [35] classified the multi-class and diagnosed different stages of kidney disease with ANN, achieving a performance of 87.70%. Polat et al. [7] conducted binary classification to predict chronic kidney disease using SVM with wrapped and filtered evaluator, achieving a performance accuracy of 98.5% with SVM filtered best first search evaluator. Mohamed et al. suggested a binary classification method to determine if a patient has CKD or not, with D-ACO providing an accuracy of 87.5% compared to other algorithms. Lakshimi et al. [23] collected data from different dialysis sites to detect kidney dialysis, analyzing it using three data mining classifier algorithms: ANN, decision tree, and logistic regression model. ANN emerged as the highest performer with 93.853% accuracy, using binary classes 'survive' and 'die' for classification, predicting patient survivability by corresponding class values. Norouzi et al. [28] addressed the Adaptive Neuro-Fuzzy Inference System (ANFIS) classifier algorithms, demonstrating its utility in diagnosing kidney failure progression over time. They exploited the binary class of kidney failure progression, achieving a prediction accuracy of 95%. Charleonnan et al. recommended an approach to predict the chronic kidney disease state of a patient with binary classes 'CKD' and 'not CKD', utilizing K-NN, SVM, decision tree, and logistic regression. However, our proposed methodology employs multi-classification and utilizes a larger dataset with more features, resulting in better performance than other works performing multi-classification. SVM provided a higher accuracy of 91.375% among 25 attributes, correctly diagnosing the chronic kidney disease stages.

3. Classification

The research focuses on chronic kidney disease (CKD), highlighting its global prevalence and significant impact on public health. To address this, the study emphasizes the importance of early detection

and intervention. Utilizing machine learning algorithms like Artificial Neural Networks (ANN), Support Vector Machine (SVM), Decision Trees (J48), Naïve Bayes (NB), Logistic Model Trees (LMT), and Functional Trees (FT), the research aims to develop accurate diagnostic tools for CKD. These algorithms enable efficient pattern recognition and classification, offering promising ways for improving patient care through timely interventions.

3.1. Artificial Neural Network (ANN)

Artificial Neural Networks and deep learning algorithms have become the state of the art in Artificial Intelligence applications for pattern recognition. ANNs are collections of a large number of neurons connected with each other in a defined way. There are numerous successful applications of ANNs in medical data to solve various problems like image analysis, drug development, interpretation, and prediction. Successful implementation of ANN-based algorithms provides more confidence to clinical practitioners as well as researchers about the achieved results. ANNs operate either in a cascaded or hierarchical way where results produced by one set of neurons are propagated to the next set. Usually, an ANN has multiple layers divided among input layer, hidden layers, and output layer. Figure 1 illustrates the general configuration of an ANN.

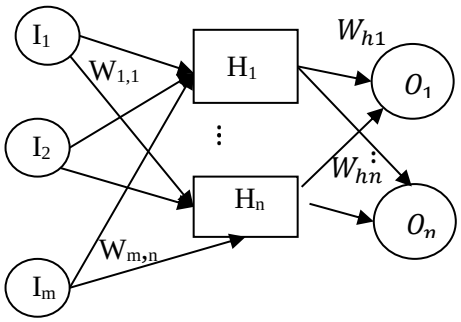

Figure 1: General ANN architecture

The input layer takes data from the environment in the form of numbers which can represent any data type like image, text, numeric or even speech. The input neurons are next connected to hidden neurons and next layer neurons (hidden layer) computes the hyper parameters for each connection as shown in the figure. The most important hyper parameters are the weights which are assigned to each link between the two layers. In figure 1, weights are denoted by W. Each next layer neuron computes the sum of products using following equation-1.

$$H_j = \sum_{i=1}^{n} I_i W_{ij} + b_j \tag{1}$$

where,

I_i = The input coming through i^{th} input neuron

J = Neuron index in hidden layer

Similarly, each output can be computed using equation-2.

$$O_j = \sum_{i=1}^{n} H_i W_{ij} + b_j \tag{2}$$

Any neuron other than input layer consists of two parts: summation and activation function that either filters or smooths the coming input. An activation function could be linear or non-linear and normally boosts the performance of classifier. ANN mostly use the supervised learning where output produced by the final layer neurons are compared with the actual target values and the difference of the predicted and actual target values, known as loss, is calculated. Based on calculated loss, the hyper-parameters (weights and biases) are fine-tuned such that they minimize the loss. The said process takes large number of iterations.

3.2. Support Vector Machine (SVM)

SVM is widely used in machine learning (ML) and pattern recognition applications due to its high performance compared to other ML methods. It is a supervised learning technique utilized for regression and classification tasks. Linear SVM supports the use of a hyperplane to separate the two classes of data. Its binary class classification capability can be readily extended for multiclass classification. SVM utilizes support vectors, which endeavor to maximize the margin from the hyperplane, as illustrated in Figure 2.

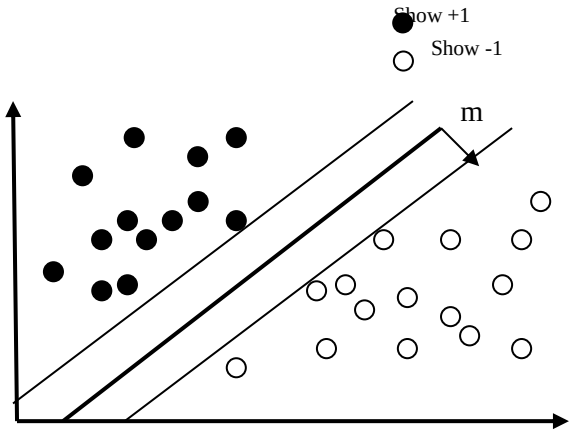

Figure 2: Linear SVM separation hyperplane

Nearest points to the margin line are called the support vector points and help in determine the optimal boundary for given classes. SVM can perform both linear and non-linear boundary detection. The negative plane represents less than 1 value in the SVM technique. It is given by equation-3.

$$z_i = m.x_i + h \leq -1 \tag{3}$$

The positive plane represents the greater than 1value in SVM technique. It is given by equation.

$$z_i = m.x_i + h \geq 1 \tag{4}$$

The classifier boundary of hyperplane is given by equation 5.

$$z_i = m.x_i + h \tag{5}$$

Nonlinearly separable hyperplane is more powerful than its linear counterpart.

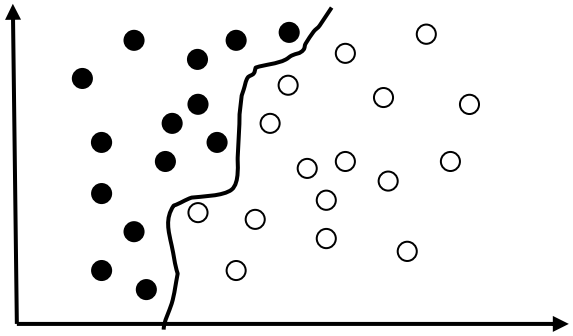

Figure 3: Non-linear SVM hyperplane

By combining both equations (3) and (4)

$$z_i(m.x_i + h) \geq 0 \ \forall_i \tag{6}$$

Quadratic Programming problem that arises during the training of SVM is solved using Sequential Minimum Optimization (SMO) algorithm [30] and optimizes the performance. SMO breaks the problem into small chunks (sub-problem), which are solved systematically one by one in a series.

3.3. J48 Decision Tree

J48 is a decision tree classifier, which is essentially a Java-based implementation of the C4.5 method. It utilizes information theory to evaluate the individual features of a given dataset. Using information gain, it determines the best split in the decision tree. The objective of J48 (Weka implementation of C4.5) is to achieve decision accuracy with flexibility. Decision tree pruning is the process that eliminates some tree nodes/branches without affecting the accuracy of the model. This removal of branches reduces the size of the tree and enhances computational efficiency. Another benefit of tree pruning is to prevent overfitting during training. J48 employs two methods of tree pruning: subtree replacement and subtree raising. Some studies have demonstrated that the pruning process can also improve the efficiency of the decision tree algorithm.

3.4. Naïve Bayesian (NB)

The Naïve Bayes algorithm is a supervised and probabilistic classifier. NB specifies conditional independence among attributes. Its simplicity lies in the simple multiplication of probabilities, reducing complexity. Due to its straightforward nature, this classification method is rapidly adopted. It achieves accurate parameter estimation by calculating the frequencies of attributes and combinations of values in a given training dataset. The algorithm has been found to be equally efficient for medical data. Despite the assumption of attribute independence, the NB classifier generates better accuracy performance. If the probability of a given data instance lying in class i is denoted by v_i, then for n classes, probabilities are given by $V = \{v\_1, v\_2, \ldots, v\_n\}$. The probability for a given instance to lie in specified classes is given by:

$$P(v_j) = \sum_{i=1}^{n} P(v_j|C_i)P(C_i) \tag{7}$$

3.5. Logistic Model Tree (LMT)

LMT is an amalgamation of the decision tree and logistic regression model. The decision tree is the most conventional method for classification, where each instance is classified based on its parameter values. It subdivides instances to determine the particular class, depending on the different values of the given parameters. Using information gain theory, the most useful attribute is selected first to define the tree. The node with the highest rank is chosen as the root of the tree. The decision tree is a subdivision of tree nodes by splitting every instance until it finds the class label. It holds a non-linear model to classify data with easy

interpretation and is preferable for a small amount of training data. Entropy D is used to measure the data impurity. For a dataset of instances $I = \{I_1, I_2, I_3, \dots, I_n\}$, entropy is given as:

$$Entropy(D) = \sum_{i=1}^{n} -I_i log I_i \tag{9}$$

And information gain for a particular feature is given by:

$$InformationGain(S) = entropy(D) - \sum_{i=1}^{m} \frac{|D_i|}{|D|} entropy D_i \tag{10}$$

The logistic regression captures the linear patterns to classify the data which portrays the data with low variance and high bias. Logistic model is preferred when there are small number of instances with noise.

Logistic regression model separates each class from parent class 'K' by using 'K-1' log odds:

$$\beta_k^T z = \log\left(\frac{P(G = k|Z = z)}{P(G = j|Z = z)}\right) \tag{11}$$

For $k = 1, \dots\dots, K-1$, β_k and P_i is the estimation of the attribute values and probability of linear function z respectively. The performance of both approaches depends on the number of instances and number of attributes of the dataset [40, 41] with no global ranking. However, LMT gives better results because it is the mixture of a logistic regression with decision tree and logistic regression is applied to the leaves of the tree using Logit Boost [42,43]. In LMT, the decision tree makes a tree that divides the set of instances into 3 regions and applies the logistic regression in every region. When both methods apply on a dataset, its performance is increased as compared to simple decision tree classifier and simple logistic regression models. LMT increases the computational complexity due to the regression function that is applied to the tree leaves.

3.6. Functional Tree

The functional tree builds the decision tree and applies logistic regression at nodes and/or leaves [44]. The decision tree is built in two phases in the functional tree algorithm. Traversing the decision tree in a depth-first tree for pruning the decision tree. In the first phase, a decision tree is built and pruning of the decision tree is done in the second phase. For pruning the tree two measures are estimated at non-leaf node.

4. Proposed Methodology

The research proposes a multiphase approach to developing a decision support system (DSS) for chronic kidney disease (CKD) diagnosis, breaking down the process into specific phases to ensure thoroughness and systematic development. It integrates data mining and machine learning techniques within the DSS framework to enhance diagnostic accuracy and performance, representing a novel approach to CKD diagnosis. Utilizing a specialized dataset sourced from a healthcare unit ensures the relevance and applicability of the data to real-world scenarios. Advanced feature ranking and selection methods prioritize high-value attributes crucial for assessing kidney function, potentially improving the DSS's accuracy. The research highlights the significance of gender-specific differences in physiological parameters for CKD diagnosis, acknowledging the importance of personalized medicine. Through experimental results analysis, the research provides insights into the performance of different classification algorithms, informing future research and clinical applications. The structure of proposed DSS is shown in figure-4.

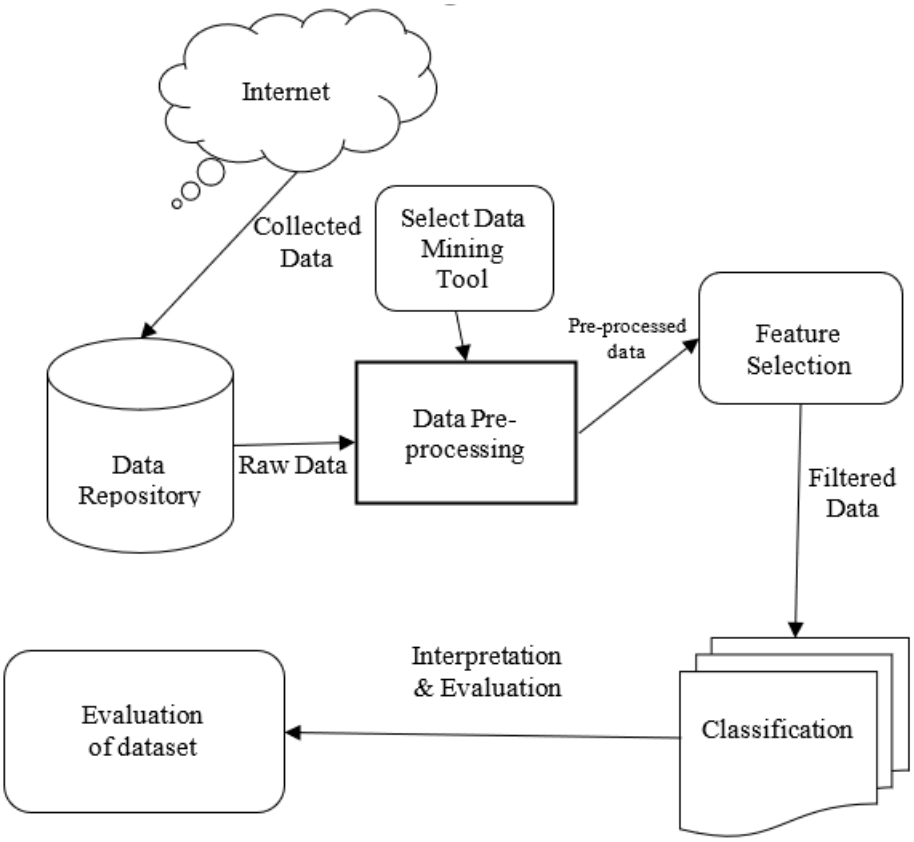

Figure 4: Flow Diagram for performance evaluation

4.1. Data collection

The data is sourced from the online machine learning repository of the University of California at Irvine, accessible at https://archive.ics.uci.edu/ml/index.php. Chronic disease dataset was collected from a healthcare unit over a period of 2 months. The dataset is provided by Dr. P. Soundarapandian, M.D., D.M., Senior Consultant Nephrologist, Apollo Hospitals, Managiri, Madurai Main Road, Karaikudi, Tamilnadu, India. It is a multivariate dataset specifically designed for classification tasks, comprising 800 instances with 25 attributes, including 11 numeric and 14 nominal attributes with multi-labels. The dataset is labeled with 5 CKD stages and one normal state. The extracted features include age, weight, gender, blood pressure, specific gravity, presence of diabetes, albumin, red blood cells, pus cells, pus cell clumps, amount of single-cell bacteria, random blood glucose, urea, serum creatinine, sodium, potassium, hemoglobin, packed cell volume, white blood cell count, hypertension, diabetes mellitus, presence of coronary artery disease, appetite, pedal edema, anemia, and the class of the instance. The five stages are CKD Stage 1: eGFR 90 or Greater, CKD Stage 2: eGFR Between 60 and 89, CKD Stage 3: eGFR Between 30 and 59, CKD Stage 4: eGFR Between 15 and 29, CKD Stage 5: eGFR Less than 15. The training dataset contains missing values and noise for some instances. Another test dataset was collected from local healthcare units. For classification, 10-fold cross-validation with a 90:10 ratio is performed on the formulated dataset. To measure the performance of classification algorithms, a separate new dataset (named TEST) is prepared by collecting relevant data from local hospitals and clinics. This test dataset has 66 instances and the same number of features.

4.2. Data mining Tool Selection

To apply the data mining algorithms and relevant data processing procedures, a data mining tool Weka 3.6 is selected. The selection is made considering its features. The tool has a number of built-in data mining

algorithms with data preprocessing features. The tool also provides the data visualization and evaluation modules.

4.3. Preprocessing

In this phase, noise and missing values are removed from dataset. Missing values are replaced with mean values. Different attributes of dataset contain values in different ranges, all such attributes are standardized in the range 0-1.

4.4Feature Ranking

The next step in diagnosis is feature ranking, where high-value features are utilized first in the mining process, while low-value features are exposed to the classifier at a later stage. For this purpose, information gain and entropy theory are employed. The highest-ranked attributes include Serum Creatinine, blood urea, hemoglobin, specific gravity, hypertension, red blood cell count, and diabetes mellitus, which manage the risk factors of kidney function. Serum Creatinine and blood urea gain high ranks because they indicate the working condition of the kidney and serve as early signs of kidney malfunction. Hemoglobin represents the rate of red blood cells in human beings, while specific gravity measures the kidney's ability to concentrate urine with plasma. Hypertension is a dangerous sign of chronic kidney disease, as it increases the risk factor of kidney failure. A smaller number of red blood cell counts in CKD patients also heightens the risk of kidney malfunction.

4.5. Classification

Classification is two step processing: feature mapping of labels and application of classification algorithm also known as statistical model. The classifier is trained using train dataset and its performance is evaluated using test/validation dataset. A number of classification methods are used in this work and their performance is reported in terms of precision, recall and f-measure. In our experiment, we first performed the binary classification and then multi-classification. The results for binary-classification are listed in table 3

Table 3: Binary class dataset results

Algorithms	Accuracy (%)	Test time (second)
LMT	98%	0.99s
FT	97.5%	0.09s
J48	99%	0.01s
ANN	99.75%	5.05s
SVM	97.75%	0.02s
Naïve Bayes	95%	0.01s

Table 3 characterizes the binary dataset classification results in terms of accuracy and test time in seconds. In binary classification, the ANN algorithm achieves 99.75% accuracy, albeit consuming more time compared to other algorithms. Naïve Bayes takes 0.01s to provide a 95% accuracy rate. Although the ANN algorithm utilizes slightly more time than other classification algorithms, it offers higher accuracy. ANN outperforms decision tree-based algorithms (LMT, FT, J48) in binary classification because the decision tree selects specific attributes from the dataset and performs classification on those attributes. However, the ANN algorithm uses all the attributes selected during preprocessing. Naïve Bayes, being a simple statistical model (non-parametric), shows lower performance as it cannot exploit the non-linearity in data at a deeper level. The decision tree (LMT, FT, J48) classifier achieves superior accuracy with insignificant differences.

In our second experiment, we conducted multi-classification for the selected dataset. Results obtained for multi-classification are listed in Table 4.

Table 4: Multi-class dataset-2 results

Algorithms	Accuracy (%)	Test Time (second)
LMT	84.5%	4.91s
FT	84.5%	0.2s
J48	84.75%	0.01s
ANN	66.75%	6.22s
SVM	67%	0.2s
Naïve Bayes	77.75%	0.1s

The J48 algorithm achieves 84.75% accuracy in 0.01 seconds. LMT and FT also yield better results with slightly lower accuracy, at 84.5% in 4.91 seconds and 0.2 seconds, respectively. Dataset-2 focuses solely on identifying CKD stages without utilizing the gender attribute. However, we observed that the gender attribute holds significant importance for CKD stages. This is because certain selected features, such as serum creatinine and hemoglobin lab reports, exhibit different normal values based on gender.

Dataset-3 includes 27 attributes, including the gender attribute. To balance the classes, an equal number of instances are added for both genders.

Table 5: Multi-class dataset-3 results

Algorithms	Accuracy (%)	Test Time (second)
LMT	91.75%	10.65s
FT	87.5%	0.49s
J48	85.875%	0.03s
ANN	76.625%	14s
SVM	75.5%	0.26s
Naïve Bayes	81.5%	0.01s

In Table 5, LMT demonstrates greater performance than SVM, primarily attributed to the inclusion of the high-ranking feature 'gender'. LMT achieves a higher accuracy than ANN as well. Specifically, LMT yields the best result with a 91.75% accuracy rate in 10.65 seconds, while SVM consumes less time (0.26 seconds) but achieves a lower accuracy of 75.5%. Based on the outcomes of the three experiments, we draw the following conclusions: In the first experiment, the results are notably high due to binary classification, which is a comparatively easier task. However, in the second experiment, the results experience a drastic drop due to the dataset's multi-classification nature. Nevertheless, in the third experiment, there is a noticeable improvement in the results, attributed to the inclusion of high-value features.

By following the outlined methodology, doctors can utilize advanced data mining and machine learning techniques to enhance their diagnostic capabilities. The DSS facilitates the integration of patient data from various sources, enabling clinicians to identify key biomarkers and clinical indicators associated with CKD progression. Through sophisticated feature ranking and selection methods, doctors can prioritize relevant attributes essential for accurate diagnosis, such as serum creatinine levels, blood pressure, and demographic factors. This systematic approach empowers healthcare professionals to make informed decisions based on evidence-driven insights extracted from patient data. Additionally, the DSS provides real-time monitoring

and predictive analytics capabilities, enabling doctors to anticipate CKD progression and implement timely interventions to mitigate adverse outcomes.

5. Evaluation and Result

Cross validation is the method to produce results with high confidence. As stated above, we used 10-fold cross validation for our experiments and we report the average score. To be surer about our results, we used another dataset for test purpose which was collected from local hospitals with the help of relevant medical practitioners.

5.1. Experiment-1

Here, we use recall, accuracy and f-measure and precision to report results of our experiments.

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{13}$$

$$f - measure = \frac{2TP}{2TP + FP + FN} \tag{14}$$

$$Preceision = \frac{TP}{TP + FP} \tag{15}$$

ANN performed best for experiment-1 and its confusion matrix is given below in table 6.

Table 6: Confusion Matrix for CKD Class

Predicted Value	**Actual**		**Performance measures**			
	Positive	**Negative**	**Recall**	**Accuracy**	**f-measure**	**Precision**
Positive	498	0	0.966	99.75%	0.998	1
Negative	2	300				

The accuracy is higher for experiment-1 and it is due to the fact that we are doing binary classification about CKD which is relatively easy problem.

5.2. Experiment-2

In experiment-2, we performed the multi-classification. In this experiment the dataset used contains all features except gender. Among all classifiers, J48 performed best and following table 8 shows its confusion matrix.

Table 7: Confusion Matrix for multi-classification

Predicted Value per CKD stage	Actual						Performance measures			
	1	**2**	**3**	**4**	**5**	**0**	**Recall**	**Accuracy**	**f-measure**	**Precision**
1	46	12	2	4	0	8	0.76	84.5	0.86	0.99
2	10	48	14	0	0	2				
3	4	16	134	6	0	2				
4	2	2	16	68	8	0				
5	0	0	0	12	84	0				
0	0	4	0	0	0	296				

The J48 algorithm achieves the best result for dataset-2 with an 84.5% performance accuracy. The description of each stage is specified as follows:

- For CKD stage 1 and CKD stage 2 patients, the age and weight differ, while the serum creatinine value remains the same.
- CKD stage 3 is determined by the serum creatinine and RBCC (Red Blood Cell Count) attribute values, with the RBCC value being less than 4.3 million/cm.
- For CKD stage 4, the serum creatinine value is greater than or equal to 1.6 mg/dl.
- The attributes values for CKD stage 5 are very close to those of CKD stage

5.3. Experiment-3

In this experiment, we added the gender attribute in the dataset and performed the multi-classification. Following table 8 shows the confusion matrix and results obtained for high performing method i.e., LMT.

Table 8: Confusion Matrix for multiclassification on dataset-3

Predicted Value per CKD stage	Actual						Performance measures			
	1	**2**	**3**	**4**	**5**	**0**	**Recall**	**Accuracy**	**f-measure**	**Precision**
1	53	7	0	0	0	0	0.868	91.75	0.93	1
2	9	60	5	1	0	1				
3	0	10	143	5	0	0				
4	0	3	12	82	3	0				
5	0	0	0	10	96	0				
0	0	0	0	0	0	300				

Table 8 shows that dataset-3 gives better performance than dataset-2 for each stage of CKD. Serum creatinine, gender, age, and weight are required to find the stages of CKD but all the other attributes that are used in the dataset are significant to predict and control the CKD stages. Serum creatinine, RBCs state, number of RBCC, anemia sign and hemoglobin attributes seen in CKD stage 5 patients in critical condition. Every stage of CKD is very close to the other.

5.4. Classification evaluation using proprietary test dataset

A separate test dataset (TEST) is used to validate the results of our trained models by using LMT algorithm. The results obtained using this dataset are given in table 9.

Table 9: Confusion Matrix for multi-classification on TEST

Predicted Value per CKD stage	**Actual**						**Performance measures**			
	1	**2**	**3**	**4**	**5**	**0**	**Recall**	**Accuracy**	**f-measure**	**Precision**
1	10	0	0	0	0	1	0.864	86.36	0.913	1
2	0	5	3	0	0	0				
3	0	1	9	0	0	2				
4	0	0	0	11	2	0				
5	0	0	0	0	12	0				
0	0	0	0	0	0	10				

By utilizing advanced data mining and machine learning techniques as outlined in the developed DSS framework, clinicians significantly enhance their diagnostic capabilities for chronic kidney disease (CKD). The experiments demonstrate the DSS's efficacy in accurately diagnosing CKD and predicting the disease stage, leveraging patient data from diverse sources. Through sophisticated feature ranking and selection methods, the DSS enables clinicians to identify crucial biomarkers and clinical indicators associated with CKD progression. Moreover, the DSS provides real-time monitoring and predictive analytics, empowering doctors to anticipate disease progression and implement timely interventions.

6. Conclusion and Future Work

In conclusion, this research proposes a comprehensive framework for the clinical decision support system (DSS) of chronic kidney disease (CKD) diagnosis and progression prediction. By utilizing high-performance data mining techniques and classification algorithms, the study aimed to improve the accuracy and efficiency of CKD diagnosis, enabling both patients and medical practitioners to make informed decisions. The findings indicate that the proposed methodology significantly enhances the diagnostic capabilities for CKD, particularly in early detection and disease progression prediction. By prioritizing high-value features and advanced machine learning techniques, the developed DSS demonstrates its potential to assist healthcare professionals in making timely and accurate diagnoses, thereby improving patient outcomes and reducing the burden of CKD-related complications. The continued refinement and validation of the proposed DSS framework, along with the exploration of emerging technologies and methodologies, hold promise for advancing the diagnosis and management of CKD and improving patient care in the future.

References

[1] Luyckx, Valerie A., Marcello Tonelli, and John W. Stanifer. "The global burden of kidney disease and the sustainable development goals." *Bulletin of the World Health Organization* 96, no. 6 (2018): 414.

[2] Versino, Elisabetta, and Giorgina Barbara Piccoli. "Chronic kidney disease: the complex history of the organization of long-term care and bioethics. Why now, more than ever, action is needed." *International Journal of Environmental Research and Public Health* 16, no. 5 (2019): 785.

[3] Pandya, Divya, Anil Kumar Nagrajappa, and K. S. Ravi. "Assessment and correlation of urea and creatinine levels in saliva and serum of patients with chronic kidney disease, diabetes and hypertension–a research study." *Journal of clinical and diagnostic research: JCDR* 10.10 (2016): ZC58.

[4] Muthulakshmi, I. "An Extensive Survey on Evolutionary Algorithm Based Kidney Disease Prediction." In *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)*, pp. 1-5. IEEE, 2019.

[5] Kumar, Manish. "Prediction of chronic kidney disease using random forest machine learning algorithm." *International Journal of Computer Science and Mobile Computing* 5, no. 2 (2016): 24-33.

[6] www.nhlbi.nih.gov/health/health-topics/topics/hbp.

[7] Polat, Huseyin, Homay Danaei Mehr, and Aydin Cetin. "Diagnosis of chronic kidney disease based on support vector machine by feature selection methods." *Journal of medical systems* 41 (2017): 1-11.

[8] Bala, Suman, and Krishan Kumar. "A literature review on kidney disease prediction using data mining classification technique." *International Journal of Computer Science and Mobile Computing* 3, no. 7 (2014): 960-967.

[9] Charleonnan, Anusorn, Thipwan Fufaung, Tippawan Niyomwong, Wandee Chokchueypattanakit, Sathit Suwannawach, and Nitat Ninchawee. "Predictive analytics for chronic kidney disease using machine learning techniques." In *2016 management and innovation technology international conference (MITicon)*, pp. MIT-80. IEEE, 2016.

[10] Rodrigues, Mariana, Hugo Peixoto, Marisa Esteves, and José Machado. "Understanding stroke in dialysis and chronic kidney disease." *Procedia computer science* 113 (2017): 591-596.

[11] Neves, José, M. Rosário Martins, João Vilhena, João Neves, Sabino Gomes, António Abelha, José Machado, and Henrique Vicente. "A soft computing approach to kidney diseases evaluation." *Journal of medical systems* 39 (2015): 1-9.

[12] Hill, Nathan R., Samuel T. Fatoba, Jason L. Oke, Jennifer A. Hirst, Christopher A. O'Callaghan, Daniel S. Lasserson, and FD Richard Hobbs. "Global prevalence of chronic kidney disease–a systematic review and meta-analysis." *PloS one* 11, no. 7 (2016): e0158765.

[13] https://www.kidney.org/news/newsroom/factsheets/KidneyDiseaseBasics#:~:text=1%20in%203%20American%20adults,lived%20with%20a%20kidney%20transplant.

[14] American Diabetes Association. "9. Cardiovascular disease and risk management: standards of medical care in diabetes—2018." *Diabetes care* 41, no. Supplement_1 (2018): S86-S104.

[15] Karthikeyan, T., and P. Thangaraju. "Analysis of classification algorithms applied to hepatitis patients." *International Journal of Computer Applications* 62, no. 15 (2013): 2530.

[16] Karthikeyan, T., and P. Thangaraju. "Best first and greedy search based CFS-Naïve Bayes classification algorithms for hepatitis diagnosis." *Biosciences and Biotechnology Research Asia* 12, no. 1 (2015): 983-990.

[17] Mittal, Ansh, Deepika Kumar, Mamta Mittal, Tanzila Saba, Ibrahim Abunadi, Amjad Rehman, and Sudipta Roy. "Detecting pneumonia using convolutions and dynamic capsule routing for chest X-ray images." *Sensors* 20, no. 4 (2020): 1068.

[18] Iraji, Mohammad Saber. "Prediction of post-operative survival expectancy in thoracic lung cancer surgery with soft computing." *Journal of Applied Biomedicine* 15, no. 2 (2017): 151-159.

[19] Kinaan, Mustafa, Hanford Yau, Suzanne Quinn Martinez, and Pran Kar. "Concepts in Diabetic Nephropathy: From Pathophysiologyto Treatment." *Journal of Renal and Hepatic Disorders* 1, no. 2 (2017): 10-24.

[20] Webster, Angela C., Evi V. Nagler, Rachael L. Morton, and Philip Masson. "Chronic kidney disease." *The lancet* 389, no. 10075 (2017): 1238-1252.

[21] Das, Himansu, Bighnaraj Naik, and H. S. Behera. "Classification of diabetes mellitus disease (DMD): a data mining (DM) approach." In *Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2017*, pp. 539-549. Springer Singapore, 2018.

[22] Mohapatra, Subasish, Prashanta Kumar Patra, Subhadarshini Mohanty, and Bhagyashree Pati. "Smart health care system using data mining." In *2018 International Conference on Information Technology (ICIT)*, pp. 44-49. IEEE, 2018.

[23] Fan, Li, Andrew S. Levey, Vilmundur Gudnason, Gudny Eiriksdottir, Margret B. Andresdottir, Hrefna Gudmundsdottir, Olafur S. Indridason, Runolfur Palsson, Gary Mitchell, and Lesley A. Inker. "Comparing GFR estimating equations using cystatin C and creatinine in elderly individuals." *Journal of the American Society of Nephrology* 26, no. 8 (2015): 1982-1989.

[24] Borisagar, Nilesh, Dipa Barad, and Priyanka Raval. "Chronic kidney disease prediction using back propagation neural network algorithm." In *Proceedings of International Conference on Communication and Networks: ComNet 2016*, pp. 295-303. Springer Singapore, 2017.

[25] Vijayarani, S., S. Dhayanand, and M. Phil. "Kidney disease prediction using SVM and ANN algorithms." *International Journal of Computing and Business Research (IJCBR)* 6, no. 2 (2015): 1-12.

[26] En Espanol, Chronic Kidney Diseases *http://www.kidney.org/kidneydisease/aboutckd.cfm*

[27] Ahmad, Mubarik, et al. "Diagnostic decision support system of chronic kidney disease using support vector machine." *2017 Second International Conference on Informatics and Computing (ICIC)*. IEEE, 2017.

[28] Norouzi, Jamshid, Ali Yadollahpour, Seyed Ahmad Mirbagheri, Mitra Mahdavi Mazdeh, and Seyed Ahmad Hosseini. "Predicting renal failure progression in chronic kidney disease using integrated intelligent fuzzy expert system." *Computational and mathematical methods in medicine* 2016, no. 1 (2016): 6080814.

[29] De Guia, Justin D., Ronnie S. Concepcion, Argel A. Bandala, and Elmer P. Dadios. "Performance comparison of classification algorithms for diagnosing chronic kidney disease." In *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pp. 1-7. IEEE, 2019.

[30] Virkar, Hemant, Karen Stark, and Jacob Borgman. "Machine learning method that modifies a core of a machine to adjust for a weight and selects a trained machine comprising a sequential minimal optimization (SMO) algorithm." U.S. Patent 9,082,083, issued July 14, 2015.

[31] Rehman, Amjad, Naveed Abbas, Tanzila Saba, Syed Ijaz ur Rahman, Zahid Mehmood, and Hoshang Kolivand. "Classification of acute lymphoblastic leukemia using deep learning." *Microscopy Research and Technique* 81, no. 11 (2018): 1310-1317.

[32] Ramzan, Farheen, Muhammad Usman Ghani Khan, Asim Rehmat, Sajid Iqbal, Tanzila Saba, Amjad Rehman, and Zahid Mehmood. "A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks." *Journal of medical systems* 44 (2020): 1-16.

[33] Rehman, Amjad, Muhammad A. Khan, Zahid Mehmood, Tanzila Saba, Muhammad Sardaraz, and Muhammad Rashid. "Microscopic melanoma detection and classification: A framework of pixel-based fusion and multilevel features reduction." *Microscopy research and technique* 83, no. 4 (2020): 410-423.

[34] Sharif, Uzma, Zahid Mehmood, Toqeer Mahmood, Muhammad Arshad Javid, Amjad Rehman, and Tanzila Saba. "Scene analysis and search using local features and support vector machine for effective content-based image retrieval." *Artificial Intelligence Review* 52 (2019): 901-925.

[35] Vijayarani, S., and S. Dhayanand. "Data mining classification algorithms for kidney disease prediction." *Int J Cybernetics Inform* 4, no. 4 (2015): 13-25.

[36] Ahishakiye, Emmanuel, Danison Taremwa, Elisha Opiyo Omulo, and Ivan Niyonzima. "Crime prediction using decision tree (J48) classification algorithm." *International Journal of Computer and Information Technology* 6, no. 3 (2017): 188-195.

[37] Quinlan, John R. "Learning with continuous classes." In *5th Australian joint conference on artificial intelligence*, vol. 92, pp. 343-348. 1992.

[38] Dimitoglou, George, James A. Adams, and Carol M. Jim. "Comparison of the C4. 5 and a Naïve Bayes classifier for the prediction of lung cancer survivability." *arXiv preprint arXiv:1206.1121* (2012).

[39] Ahmed, Fahim, and Kyoung-Yun Kim. "Data-driven weld nugget width prediction with decision tree algorithm." *Procedia Manufacturing* 10 (2017): 1009-1019.

[40] Huang, Tingkai, Bingchan Li, Dongqin Shen, Jie Cao, and Bo Mao. "Analysis of the grain loss in harvest based on logistic regression." *Procedia computer science* 122 (2017): 698-705.

[41] Kazakevičiūtė, Agne, and Malini Olivo. "Point separation in logistic regression on Hilbert space-valued variables." *Statistics & Probability Letters* 128 (2017): 84-88.

[42] Patel, Jaymin, Dr TejalUpadhyay, and Samir Patel. "Heart disease prediction using machine learning and data mining technique." *Heart Disease* 7, no. 1 (2015): 129-137.

[43] Xing, Chao, Xin Geng, and Hui Xue. "Logistic boosting regression for label distribution learning." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4489-4497. 2016.

[44] Siraj, Fadzilah, and Mansour Ali. *Mining enrollment data using descriptive and predictive approaches*. InTech—Open Access Company, 2011.

[45] Thiyagaraj, M., and G. Suseendran. "Research of Chronic Kidney Disease based on Data Mining Techniques." *International Journal of Recent Technology and Engineering (IJRTE). ISSN* (2019): 2277-3878.

[46] Elhoseny, Mohamed, K. Shankar, and J. Uthayakumar. "Intelligent diagnostic prediction and classification system for chronic kidney disease." *Scientific reports* 9, no. 1 (2019): 9583.

[47] Sobrinho, Alvaro, Andressa CM Da S. Queiroz, Leandro Dias Da Silva, Evandro De Barros Costa, Maria Eliete Pinheiro, and Angelo Perkusich. "Computer-aided diagnosis of chronic kidney disease in developing countries: A comparative analysis of machine learning techniques." *IEEE Access* 8 (2020): 25407-25419.

[48] Davenport, Thomas, and Ravi Kalakota. "The potential for artificial intelligence in healthcare." *Future healthcare journal* 6, no. 2 (2019): 94-98.

[49] Bhargava, Neeraj, Girja Sharma, Ritu Bhargava, and Manish Mathuria. "Decision tree analysis on j48 algorithm for data mining." *Proceedings of international journal of advanced research in computer science and software engineering* 3, no. 6 (2013).

Machines and Algorithms
http://www.knovell.org/mna

Review Article

Cloud Computing Advances and Architectures

Alishba Kamran[1,*], Anas Tanvir[1], Usama Bin Imran[1] and Danyal Farhat[1]

[1]Department of Data Science, FAST National University of Computer and Emerging Sciences, 54770, Lahore, Pakistan

[*]Corresponding Author: Alishba Kamran. Email: l216297@lhr.nu.edu.pk

Received: 20 April 2023; Revised: 4 June 2023; Accepted:18 July 2023; Published: 16 August 2023

AID: 002-02-000023

Abstract: Cloud computing has emerged as a transformative technology, revolutionizing the technological landscape with its scalability, flexibility, and cost-effectiveness. However, its rapid evolution has introduced challenges in resource management, efficiency, and security. This study aims to explore recent advancements in cloud computing architectures to bridge research gaps and propel the industry forward. Through an exhaustive literature study, data synthesis, meticulous examination, and presentation of findings, this research comprehensively analyzes recent developments in cloud computing. It also includes experimental setups to evaluate load balancing mechanisms, resource allocation algorithms, and data mining techniques. The study demonstrates significant improvements in resource efficiency, reduced response times, and enhanced scalability within cloud systems. Advanced methodologies in load balancing, resource allocation, and data mining contribute to optimizing cloud infrastructure performance. The findings underscore the transformative potential of innovative methodologies in cloud computing architectures. Future research directions include exploring advanced data mining techniques, enhancing load balancing mechanisms, and addressing privacy and security challenges. These endeavors will drive innovation, improve reliability, and ensure the continued evolution of cloud computing technology.

Keywords: Cloud computing; architecture; load balancing; resource allocation; data mining; energy efficiency; privacy; security;

1. Introduction

In recent years, cloud computing has risen to prominence, reshaping the technological landscape with its unparalleled scalability, flexibility, and cost-effectiveness. This transformative technology has not only revolutionized the way computing resources and services are delivered but has also empowered organizations and individuals alike. By providing on-demand access to processing power and the ability to swiftly scale activities, cloud computing has become an indispensable tool in the modern era.

However, the rapid evolution of cloud environments has introduced a myriad of new challenges. From ensuring effectiveness and security to efficiently allocating resources, the development of cloud technologies has brought about a host of complexities that must be addressed. As such, this study endeavors to explore the latest advancements in cloud computing architectures, with the overarching goal of bridging identified research gaps and propelling the industry forward into the future.

As cloud computing continues to experience rapid expansion, the need to address critical issues related to resource management, efficiency, and security has never been more pressing. As businesses across diverse industries increasingly rely on cloud-based solutions to drive innovation and streamline operations, staying abreast of the latest advancements in cloud computing architectures is imperative. Through the elucidation of recent developments and a comprehensive discussion of current research gaps, this paper aims to equip stakeholders with the requisite information and understanding to navigate the intricate terrain of cloud computing effectively.

Given the ever-changing nature of technology, it is paramount to take a proactive stance in addressing the complex challenges inherent in cloud computing settings. By leveraging the most recent developments in cloud computing infrastructures, organizations can bolster their defenses against emerging threats and unearth new avenues for innovation. Thus, this article meticulously examines research flaws and seminal breakthroughs in cloud technology, laying the groundwork for a roadmap towards a more secure, efficient, and resilient cloud computing ecosystem.

2. Research Methodology

In this section, we embark on a detailed exploration of the methodology employed to investigate current advancements in cloud computing architectures and address identified gaps in the existing research. Additionally, we elucidate the scope of our work to provide clarity on the breadth and depth of our study.

Our research methodology is designed to ensure comprehensive coverage of the latest developments in cloud computing architectures and environments.

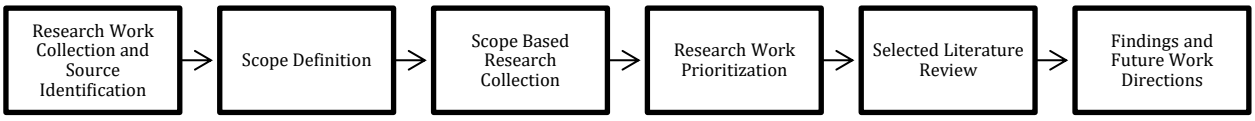

Figure 1: Review Research Methodology

This initial phase involves gathering all relevant research work and identifying credible sources like google scholar and IEEE explore. The goal is to compile a comprehensive set of materials, such as academic papers, articles, reports, and other publications, that are pertinent to the research topic.

In this phase, the scope of the research is clearly defined. This involves specifying the boundaries and focus areas of the study, including the key questions to be answered, and the specific aspects of the topic to be examined. Defining the scope helps to narrow down the vast amount of available research to what is most relevant to the study's objectives. We have identified five major areas in this phase and then our work proceeds in that direction

Once the scope is defined, the next step is to collect research works that fall within this defined scope. This involves filtering the initial collection of sources to include only those that directly address the research areas and fit within the specified boundaries. The purpose is to ensure that the research is focused and manageable.

In 4^{th} phase, the collected research works are prioritized based on their relevance, quality, and importance to the study. Criteria for prioritization includes the credibility of the source, the methodology used, the impact of the findings, and how closely the research aligns with the study's objectives.

The 5^{th} phase is core of this work where after prioritizing the research works, a detailed review of the selected literature is conducted. The literature review provides a comprehensive understanding of the current state of knowledge on the topic and forms the foundation for further research.

The final phase involves summarizing the findings from the literature review and identifying directions for future research. This includes highlighting the main conclusions drawn from the existing studies, noting

any unresolved issues, and proposing areas where further investigation is needed. This phase helps to outline the contribution of the research and suggest pathways for future studies to build upon.

2.1. Scope of the Work

The scope of our study encompasses an exploration of cloud computing's latest developments and architectural issues. Our focus extends to pivotal topics such as load balancing systems, resource allocation algorithms, energy-efficient resource management, and privacy and security concerns within cloud computing infrastructures. Leveraging information from credible sources, our research endeavors to offer insightful suggestions and recommendations to industry stakeholders, thereby fostering informed decision-making and driving innovation in the realm of cloud computing.

3. RELATED WORK

In the realm of cloud computing, several important topics have been explored by researchers to enhance the efficiency and security of cloud-based systems. In this context, we review five important areas which play critical role in cloud computing. Let's delve into some key areas of research in this field:

3.1. Load Balancing Mechanisms in Cloud Computing Environments

Recent research has explored the field of load balancing in cloud computing, highlighting various aspects and challenges in this area. Mohammadian et al. [7] conducted a systematic literature review on fault-tolerant load balancing in cloud computing, providing insights into existing tools and methods. With the increasing number of cloud users and their requests, cloud systems can become either underloaded or overloaded, leading to issues like high response times and power consumption. Load balancing methods are crucial for addressing these problems and improving cloud server performance by distributing the load among nodes. Over the past decade, this challenge has garnered significant research interest, resulting in various proposed solutions. This paper presents a detailed and systematic review of fault-tolerant load balancing methods in cloud computing. Existing methods are identified, classified, and analyzed based on qualitative metrics like scalability, response time, availability, throughput, reliability, and overhead. Additional criteria considered include whether the methods use a dynamic or static approach, heuristic or meta-heuristic techniques, reactive or proactive fault tolerance, simulation tools, and the types of detected faults. A comparative analysis of these methods is provided, along with an examination of challenges, research trends, and open issues. The study finds that static methods, which require prior knowledge of the system's status, are less effective in resource utilization and reliability compared to dynamic methods. Dynamic methods, which can manage varying load conditions, offer better performance but pose challenges in algorithm development for dynamic cloud environments.

In another recent work [8], Sundas et al. explored modified bat algorithms for optimal virtual machines in cloud computing, focusing on load balancing and service brokering techniques. The authors discuss how to calculate Fitness Values (FCVs) to provide reliable solutions for load balancing during its initial stages in a cloud computing environment, which includes both physical and logical components such as cloud infrastructure, storage, services, and platforms. The study focuses on load balancing, introducing a new approach for balancing loads among Virtual Machines (VMs). The proposed method is based on a Modified Bat Algorithm (MBA), which operates in two variants: MBA with Overloaded Optimal Virtual Machine (MBAOOVM) and MBA with Balanced Virtual Machine (MBABVM). The MBA generates cost-effective solutions, and its strengths are validated by comparing it with the original Bat Algorithm. The authors have run their proposed algorithm for 500 iterations with bat population equal to 50 for various bench marks. he modified bat algorithm enhances performance in quality of service (QOS), energy management, resource scheduling, and load balancing. The research suggests that hybridizing the bat algorithm with other meta-heuristic techniques could further enhance its performance and applicability to other fields.

Challenges faced during load balancing in cloud computing are studied by Sultan & Khaleel in [9], emphasizing the importance of load balancing in addressing system setup and operational issues. Load balancing algorithms (LBAs) are crucial for distributing workloads evenly and ensuring consistent service

quality. The paper discusses various LBAs that enhance resource utilization and better meet user needs across multiple parameters. It highlights the importance of load balancing in managing storage, on-demand services, and data centers. Effective load balancing reduces server overhead, maximizes resource usage and throughput, minimizes processor migration time, and improves overall system performance and efficiency. In [11] Oduwole et al. offered a retrospective view on cloud computing load balancing techniques, highlighting the limitations of current strategies. Furthermore, a comprehensive survey on recent load balancing techniques in cloud computing was presented in the International Journal of Advanced Trends in Computer Science and Engineering in 2021, emphasizing the significance of load balancing in maximizing output and resource utilization while minimizing costs. To address the issues associated with load balancing, a central-distributive framework based on throughput maximization is proposed. This framework includes a central data center (DC) and up to five regional DCs. User requests are handled by the regional DCs where they originate, with load balancing occurring at two levels: Level 1 (regional) using Particle Swarm Optimization (PSO) and Level 2 (central) using the Firefly method. Tasks are categorized into Group A (handled locally) and Group B (handled centrally). The PSO algorithm is preferred for regional load balancing due to its efficiency in finding optimal solutions, while the Firefly method is used at the central level for its high processing speed. This system reduces the need to transfer tasks, thereby lowering response time and costs while increasing throughput by prioritizing tasks that maximize throughput.

Tawfeeg et al. [10] conducted a systematic literature review on cloud dynamic load balancing and reactive fault tolerance techniques, focusing on the relationship between these techniques in cloud environments. This paper systematically reviews reactive fault tolerance, dynamic load balancing, and their integration. The comparative analysis revealed that combining reactive fault tolerance and dynamic load balancing can enhance availability and reliability. Current frameworks often address limited fault types, but advancements in machine learning, including deep learning algorithms, can improve fault detection and resource management. Hybrid fault tolerance techniques like replication, checkpointing, and migration are discussed. Most existing fault tolerance approaches focus on reactive techniques, emphasizing the need for comprehensive frameworks that include prediction, detection, prevention, and recovery. Meanwhile, dynamic load balancing frameworks often overlook load imbalances, potentially leading to service denials. Incorporating deep learning algorithms can improve load balance and performance. The integration of reactive fault tolerance with dynamic load balancing can address hardware and software failures. Effective load distribution and clustering techniques can enhance job-node mapping and fault tolerance in distributed networks. Moreover, Rani et al. (2022) explored state-of-the-art dynamic load balancing algorithms for cloud computing, aiming to optimize performance parameters based on different load balancing systems.

Table 1: Load Balancing Mechanisms in Cloud Computing Environments

Load Balancing Mechanism	Description	Advantages	Disadvantages	Examples
Round Robin	Distributes incoming requests sequentially across servers.	Simple to implement, ensures even distribution if all servers have similar capacity.	Doesn't consider server load, can overload less powerful servers.	Nginx, HAProxy
Least Connection	Directs traffic to the server with the fewest active connections.	Efficient for environments with long-lived connections, balances load based on actual traffic.	Can lead to uneven load if connection durations vary significantly.	Apache, HAProxy
Weighted Round Robin	Similar to Round Robin but assigns weights to servers based on their	Takes server capacity into account, better load distribution for heterogeneous	More complex to configure, weight determination can be challenging.	Nginx, HAProxy

	capacity.	environments.		
Resource-Based	Balances load based on specific resource usage (CPU, memory, etc.).	Optimizes resource utilization, prevents overload.	Requires monitoring of server metrics, more complex implementation.	Azure Load Balancer, Amazon ELB
IP Hash	Uses the client's IP address to determine which server will handle the request.	Ensures that a client consistently connects to the same server, useful for session persistence.	Can lead to uneven distribution if IPs are not evenly distributed.	Nginx, F5 Big-IP
Random	Assigns incoming requests to servers randomly.	Simple to implement, avoids bias in request assignment.	Doesn't account for server load or capacity, can lead to suboptimal performance.	Custom implementations
Dynamic Load Balancing	Adjusts load distribution based on real-time performance metrics and changing conditions.	Highly efficient, adapts to changing workloads and server states.	High complexity, requires continuous monitoring and adjustment.	Kubernetes, Traefik
Geographic Load Balancing	Directs traffic based on the geographic location of the client.	Reduces latency by routing to the nearest server, improves user experience.	Requires a global network of servers, can be complex to manage.	Cloudflare, AWS Global Accelerator
Application-Aware Load Balancing	Considers the specific needs of applications (e.g., CPU-intensive vs. memory-intensive tasks).	Optimizes performance based on application characteristics, improves resource utilization.	High complexity, requires deep understanding of application requirements.	NGINX Plus, Citrix ADC

3.2. Resource Allocation Algorithms for Cloud Data Centers

A detailed evaluation of resource allocation algorithms for virtualized cloud data centers is discussed. In [13], authors provide an in-depth understanding of resource allocation in cloud computing, identifying gaps between existing techniques and areas requiring further investigation. It categorizes 77 research papers from 2007 to 2020, offering a taxonomy and summarizing key developments in resource allocation techniques over these years. The article highlights promising future directions, emphasizing the need for more cost-effective allocation schemes. Future focus areas should include enhancing security, performance isolation, smooth virtual machine migration, interoperability, resilience to failure, graceful recovery, and reducing data center operational costs. The study predicts that cloud computing services will become integral to various information systems of all scales.

Seyed Majid Mousavi et. al. in [14] examine the performance of two relatively new optimization algorithms, TLBO and GW, as well as a hybrid of these algorithms, in dynamic resource allocation. Experimental results comparing the hybrid algorithm with TLBO and GW show that the hybrid approach performs more efficiently than either algorithm alone. The study concludes that the primary challenge in cloud scheduler resource allocation is the lack of convergence to an optimal solution. Optimizing objective functions for resource allocation in real-time poses a significant challenge. Evaluation of experimental results demonstrates that the proposed hybrid approach outperforms the other methods, particularly in high-volume data scenarios.

Another work related to resource allocation [15] introduces an enhanced optimization algorithm for resource allocation, aiming to minimize deployment costs while enhancing Quality of Service (QoS)

performance. This algorithm accommodates diverse customer QoS needs within budget constraints. Experimental analysis, performed by deploying various workloads on Amazon Web Services, demonstrates the effectiveness of the proposed algorithm.

Table 2: Popular Resource Allocation Algorithms for Cloud Data Centers

Algorithm	Description	Advantages	Disadvantages	Examples/Use Cases
First-Come, First-Served (FCFS) [12]	Allocates resources in the order requests are received.	Simple to implement and understand.	Can lead to poor resource utilization and performance under heavy load.	Basic scheduling in small cloud environments.
Round Robin [3]	Allocates resources in a circular order to ensure fairness.	Simple and fair, prevents starvation.	Doesn't consider job length or priority, can lead to inefficient resource usage.	Load balancing in web servers.
Ant Colony Optimization (ACO) [6]	Uses the behavior of ants to find optimal paths for resource allocation.	Efficient for complex problems, adaptable to changes.	High computational cost, convergence time can be long.	Network routing, job scheduling in cloud environments.
Simulated Annealing (SA) [5]	Uses probabilistic techniques to find an optimal solution by exploring the search space.	Good at avoiding local optima, can handle large search spaces.	Slow convergence, parameter tuning required.	Energy-efficient resource scheduling.
Dynamic Resource Allocation (DRA) [4]	Adjusts resources in real-time based on current demand and workload.	High adaptability, improves resource utilization and performance.	Complex to implement, requires continuous monitoring.	Autoscaling in cloud environments like AWS, Azure.
Multi-Objective Optimization (MOO)	Balances multiple objectives such as cost, performance, and energy efficiency.	Can provide balanced solutions considering various factors.	Computationally intensive, complex to solve.	Energy-efficient cloud computing, QoS management.
Task Consolidation Algorithms	Consolidates tasks to reduce the number of active servers, saving energy.	Improves energy efficiency, reduces operational costs.	Can lead to resource contention, performance degradation.	Green cloud computing, energy-efficient data centers.
Resource Prediction Algorithms	Predicts future resource needs based on historical data and trends.	Helps in proactive resource management, improves utilization.	Accuracy depends on prediction model, requires historical data.	Predictive autoscaling, capacity planning.

3.3. Scalability and Elasticity

These are two fundamental concepts in cloud computing that help organizations efficiently manage and optimize their resources to meet varying demands. Scalability refers to the ability of a system, network, or process to handle a growing amount of work or its potential to accommodate growth. Elasticity refers to the ability of a system to automatically adjust the resources allocated to it in response to changes in demand. Perri D. et all. [16] explore the potential of cloud containers and provides guidelines for companies and organizations on migrating legacy infrastructure to a modern, reliable, and scalable setup. Cloud containers are lightweight, portable units that package an application and its dependencies, enabling the application to

run consistently across different computing environments. Containers facilitate rapid infrastructure expansion and increased processing capacity. The work proposes an architecture based on the "Pilot Light" topology, which balances cost and benefits. Services are reconfigured into small Docker containers, with workload balanced using load balancers to enable future horizontal autoscaling. This approach allows for the generation of additional containers, helping companies model and calibrate their infrastructure based on user projections. The proposed method results in a maintainable and fault-tolerant system, particularly beneficial for small and medium-sized organizations. The Pilot Light model ensures long-term reliability and minimal data loss (low Recovery Point Objectives (RPO) and Recovery Time Objective (RTO)) during issues like hacker attacks or natural disasters. Their future plans include offering pre-configured Docker images and using the Infrastructure as Code (IAAC) paradigm to describe and automate virtual structures across organizations.

In [17] Sehgal Nk et. all. explore the factors driving changes in demand, such as the rise of remote work and telemedicine. Based on these requirements, they analyze the demand of new protocols and architectures to satisfy customer latency expectations. In addition to this they provide a comprehensive review about scalable machine learning models in the cloud and cost optimization strategies for managing growth and scaling in the cloud.

Function as a Service (FaaS) is a new software technology that offers features like automated resource management and auto-scaling. However, because these operations are transparent, software engineers may not fully grasp the scaling characteristics and limitations, potentially leading to poor performance. To address this, authors in [18] conducted a study on the scalability of FaaS under intensive workloads across three major FaaS platforms: Amazon AWS Lambda, IBM, and Azure Cloud Function. They also investigated a workload smoother design pattern to see if it improves overall FaaS performance. Although the results obtained indicate that different FaaS platforms use distinct scaling strategies, however all platforms effectively auto-scale by adding resources during intensive workloads, thereby increasing system capacity. and by applying a workload smoother, software engineers can achieve success rates of 99-100%, compared to 60-80% when the FaaS system is saturated. This improvement highlights the need for a request queue with configurable options to prevent throttling issues, a feature that AWS Lambda and IBM Cloud Function should consider incorporating to enhance performance for their users.

Researchers are developing various tools to address the scalability issues for cloud environments. Liu XY et. al. in [19] introduces ElegantRL-podracer, a scalable and elastic library for cloud-native DRL, capable of supporting millions of GPU cores for massively parallel training. The requirement for such highly concurrent libraries generates due to various applications of deep reinforcement learning (DRL) like game playing and robotic control. Adopting a cloud-native approach to train DRL agents on GPU cloud platforms offers a promising solution for data collection from agent-environment interactions. ElegantRL-podracer uses a tournament-based ensemble scheme to manage training on hundreds or thousands of GPUs, coordinating interactions between a leaderboard and a training pool with hundreds of pods. This approach ensures scalability, efficiency, and accessibility in DRL training. The authors have made the code available on GitHub.

Table 3: Scalability and Elasticity

Algorithm	Description	Advantages	Disadvantages	Examples/Use Cases
Cloud Bursting [2]	Extends on-premises resources to the cloud during peak demands.	Cost-effective, handles peak loads efficiently.	Requires hybrid cloud setup, potential latency issues.	Retail during holiday seasons, financial modeling.
Serverless Architecture (Function as a Service - FaaS) [1]	Automatically scales resources based on the execution of individual functions.	Fine-grained scaling, reduced management overhead.	Cold start latency, vendor lock-in issues.	Event-driven applications, microservices.

3.4. Energy-Efficient Resource Management in Cloud Computing

Optimizing the allocation and utilization of resources in a way that minimizes energy consumption while still meeting performance requirements is known as energy efficient resource management. Energy-efficient resource management aims to achieve a balance between performance and energy consumption, ultimately reducing operational costs and environmental impact while maintaining service quality. Researchers are exploring various means of energy efficient management in cloud computing. The list of these means includes the algorithm optimization, optimized resource allocation, power management, workload scheduling, resource virtualization, and real time monitoring based optimizations.

Hussain M and Wei LF et. al. in [20] propose the Energy and Performance-Efficient Task Scheduling Algorithm (EPETS) for heterogeneous virtualized clouds to address energy consumption concerns. The proposed algorithm comprises two stages: initial scheduling prioritizes task completion within deadlines, while the second stage focuses on task reassignment to minimize energy usage within the deadline. The authors also propose an energy-efficient task priority system to strike a balance between scheduling and energy savings. Simulation results demonstrate that proposed algorithm compared to existing methods like RC-GA, AMTS, and E-PAGA, EPETS significantly reduces energy consumption while improving performance, ensuring deadline compliance.

Qunsong Zeng et al. [21] introduced a technique called computation-and-communication resource management C^2RM for edge machine learning (EML), which could lead to more energy-efficient cloud computing. EML is the deployment of learning algorithms at the network edge to train AI models using enormous distributed data and computation resources. The C^2RM architecture allows for multi-dimensional control, such as bandwidth allocation, CPU-GPU workload partitioning, speed scaling at each device, and C^2 time division per connection. The proposed framework's central component is a set of energy rate equilibriums with regard to various control variables that have been demonstrated to exist among devices or between processing units inside each device. The results are used to create efficient methods for computing optimal C^2RM policies faster than current optimisation tools. Based on the equilibriums, authors offer energy-efficient techniques for device scheduling and greedy spectrum sharing, scavenging "spectrum holes" caused by heterogeneous C^2 time divisions among devices. Experiments with a real dataset show that C^2RM improves the energy efficiency of a federated edge learning (FEEL) system.

Recently Artificial Intelligence (AI) has found its way in all type of computations. Cloud computing has no exception in this context. Various researchers, in order to optimize their proposed methods, have used AI algorithms. Zong Q, et. al. [21] claims that existing approaches, such as traditional heuristics and reinforcement learning algorithms, only partially address scalability and adaptability challenges. They frequently ignore the relationships between host thermal parameters, task resource consumption, and scheduling decisions, resulting in poor scalability and increasing compute resource requirements, particularly in contexts with changing resource demands. To address these limitations, the authors

suggested HUNTER, an AI-based comprehensive resource management technique for sustainable cloud computing. HUNTER approaches energy efficiency optimisation in data centres as a multi-objective scheduling issue, taking into account energy, thermal, and cooling models. They employed a Gated Graph Convolution Network called HUNTER to approximate Quality of Service (QoS) for system states and make optimal scheduling decisions. Experiments conducted on simulated and physical cloud environments using the CloudSim and COSCO frameworks show that HUNTER outperforms state-of-the-art baselines in terms of energy consumption, SLA violation, scheduling time, cost, and temperature, with gains of up to 12%, 35%, 43%, 54%, and 3%, respectively.

3.5. Privacy and Security Issues in Cloud Computing

The measures and protocols designed to protect data, applications, and services from unauthorized access, breaches, and other cyber threats are categorized as privacy and security issues in cloud computing. Given the nature of cloud environments, which involve storing and processing data on remote servers accessed via the internet, ensuring security and privacy is critical. A recent survey focusing on security and privacy issues published by Abdulsalam YS, Hedabou M. [22] provides a comprehensive review. The authors identify that outsourcing data and applications to the cloud introduces significant security and privacy concerns, which are critical to cloud adoption. Various security strategies have been proposed in the literature to address these concerns, along with comprehensive reviews of related issues. Despite this, existing research often lacks the flexibility to mitigate multiple threats without conflicting with cloud security objectives and fails to provide adequate technical solutions to these threats. This paper addresses these gaps by introducing adaptive solutions that align with current and future cloud security needs. Using the STRIDE approach, the authors have analyzed security threats from a user perspective and critiques inefficient solutions in the literature, offering recommendations for creating a secure, adaptive cloud environment.

Another study presented by Abba Ari AA et. al. [23] that analyzes the security issues in cloud of things (CoT). The integration of Cloud Computing (CC) and the Internet of Things (IoT) is known as the Cloud of Things (CoT) that has revolutionized ubiquitous computing. This integration is essential because IoT devices generate vast amounts of data that require CC for storage and processing. However, CoT faces significant security and privacy challenges as users and IoT devices share computing and networking resources remotely. This paper examines these issues by exploring the CoT architecture and existing applications. The study identifies and discusses various security and privacy concerns, potential challenges, and open issues that need to be addressed to ensure the safe and efficient use of CoT.

A specialized case study on security and privacy in cloud computing is published by Sajid Habib Gill et. al. [24]. The authors state that current cloud services often lack sufficient security and reliability. This research provides an in-depth analysis of the privacy and security challenges in cloud computing and underscores their importance with a case study on smart campus security using Blockchain technology. This study aims to encourage further research into cloud security issues.

The researchers are attempting to develop new strategies and algorithms to improve security and privacy. Shen J. et. al [25] proposed a privacy-preserving and untraceable scheme for multiparty data sharing using proxy re-encryption and oblivious random-access memory (ORAM). This scheme supports multiple users sharing data securely in the cloud. Group members and a proxy exchange keys during the key exchange phase, enabling them to resist multiparty collusion. The proxy re-encryption phase allows group members to implement access control and securely store data, facilitating secure data sharing. For cloud privacy, a comprehensive model is presented by Akremi A, and Rouched M [26]. The guidelines presented by Akremi A. can be followed by cloud privacy researchers to design more secure algorithms. Based on our review, we found following main issues associated with cloud security and privacy.

Table 4: Privacy and Security Issues in Cloud Computing

Issue	Description	Implications	Mitigation Strategies
Data Breaches	Unauthorized access to sensitive data stored in the cloud.	Loss of sensitive information, legal consequences, financial losses, damage to reputation.	Encryption, multi-factor authentication, regular security audits, access control policies.
Data Loss	Accidental deletion, corruption, or unavailability of data in the cloud.	Permanent loss of important data, business disruption, financial impact.	Regular data backups, data replication, disaster recovery plans, data integrity checks.
Insider Threats	Malicious actions by employees or other insiders with access to cloud resources.	Data theft, unauthorized data manipulation, service disruption.	Strict access controls, employee monitoring, security training, role-based access control (RBAC).
Denial of Service (DoS) Attacks	Overwhelming cloud services with excessive traffic, making them unavailable to legitimate users.	Service downtime, financial losses, damage to reputation.	Traffic filtering, rate limiting, scalable infrastructure, DDoS protection services.
Account Hijacking	Compromise of user accounts through phishing, password guessing, or credential theft.	Unauthorized access to cloud resources, data theft, service misuse.	Strong password policies, multi-factor authentication, anomaly detection, session management.
Data Privacy	Inadequate protection of personal or sensitive information in the cloud.	Violation of privacy regulations, legal penalties, loss of user trust.	Data anonymization, encryption, privacy impact assessments, compliance with privacy laws (e.g., GDPR).
Insecure APIs	Vulnerabilities in cloud service APIs that can be exploited by attackers.	Unauthorized access, data breaches, service disruptions.	Secure API development practices, regular security testing, API access control, use of API gateways.
Shared Technology Vulnerabilities	Security flaws in the underlying shared infrastructure (e.g., hypervisors, virtual machines).	Cross-tenant attacks, data leakage, system compromise.	Regular patching, isolation mechanisms, security configurations, continuous monitoring.
Lack of Compliance	Failure to adhere to industry regulations and standards for data security and privacy.	Legal penalties, loss of business opportunities, damage to reputation.	Compliance audits, adherence to standards (e.g., ISO 27001, SOC 2), regulatory compliance checks.
Lack of Control and Visibility	Limited visibility and control over data and operations in the cloud environment.	Difficulty in managing security, potential for undetected breaches.	Security monitoring tools, centralized management, logging and auditing, Service Level Agreements (SLAs).

4. Conclusion and Future Directions

In conclusion, this research has provided a comprehensive review of recent developments in cloud computing architectures and outlined a research approach aimed at addressing relevant research gaps in the field. Through the literature review we outlined some future work directions. One promising avenue for future research involves the investigation of advanced data mining techniques within cloud computing contexts. By leveraging cutting-edge data analytics algorithms and machine learning models, researchers can unlock new insights from large-scale cloud datasets, enabling more sophisticated analysis and decision-

making processes. These advanced data mining techniques have the potential to revolutionize resource management, predictive analytics, and anomaly detection within cloud environments, paving the way for more intelligent and data-driven cloud infrastructures.

Additionally, there is a pressing need to improve load balancing mechanisms in cloud computing to address evolving workload dynamics and optimize resource utilization. Future research efforts should focus on developing adaptive and dynamic load balancing algorithms that can efficiently distribute workloads across heterogeneous cloud resources while accounting for factors such as resource availability, performance requirements, and cost considerations. By enhancing load balancing mechanisms, organizations can achieve better resource allocation, improved scalability, and enhanced fault tolerance in cloud environments.

Furthermore, as cloud computing continues to proliferate across various industries and domains, the need to address new privacy and security challenges becomes paramount. Future research endeavors should prioritize the development of robust privacy-preserving techniques, encryption protocols, and intrusion detection systems tailored to the unique characteristics of cloud architectures. By strengthening privacy and security measures, cloud service providers can instill greater trust and confidence among users, fostering widespread adoption and sustainable growth of cloud computing technologies.

In summary, future advancements in cloud computing will hinge on the exploration of advanced data mining techniques, the enhancement of load balancing mechanisms, and the mitigation of emerging privacy and security threats. By pursuing these avenues of research, we can unlock new opportunities for innovation, improve the reliability and performance of cloud infrastructures, and ensure the continued evolution of cloud computing as a transformative technology in the digital era.

To expedite the review and typesetting process, authors must follow the Microsoft Word template provided for preparing their manuscripts. This template must be strictly adhered to when formatting the manuscript for submission.

References

[1] Shafiei, Hossein, Ahmad Khonsari, and Payam Mousavi. "Serverless computing: a survey of opportunities, challenges, and applications." *ACM Computing Surveys* 54, no. 11s (2022): 1-32.

[2] Syed, Hassan Jamil, Abdullah Gani, Raja Wasim Ahmad, Muhammad Khurram Khan, and Abdelmuttlib Ibrahim Abdalla Ahmed. "Cloud monitoring: A review, taxonomy, and open research issues." *Journal of Network and Computer Applications* 98 (2017): 11-26.

[3] Mishra, Ratan, and Anant Jaiswal. "Ant colony optimization: A solution of load balancing in cloud." *International Journal of Web & Semantic Technology* 3, no. 2 (2012): 33.

[4] Lorido-Botran, Tania, Jose Miguel-Alonso, and Jose A. Lozano. "A review of auto-scaling techniques for elastic applications in cloud environments." *Journal of grid computing* 12 (2014): 559-592.

[5] Zhang, Qi, Lu Cheng, and Raouf Boutaba. "Cloud computing: state-of-the-art and research challenges." *Journal of internet services and applications* 1 (2010): 7-18.

[6] Li, Kun, Gaochao Xu, Guangyu Zhao, Yushuang Dong, and Dan Wang. "Cloud task scheduling based on load balancing ant colony optimization." In *2011 sixth annual ChinaGrid conference*, pp. 3-9. IEEE, 2011.

[7] Mohammadian, Vahid, Nima Jafari Navimipour, Mehdi Hosseinzadeh, and Aso Darwesh. "Fault-tolerant load balancing in cloud computing: A systematic literature review." *IEEE Access* 10 (2021): 12714-12731.

[8] Sundas, Amit, Sumit Badotra, Youseef Alotaibi, Saleh Alghamdi, and Osamah Ibrahim Khalaf. "Modified Bat Algorithm for Optimal VM's in Cloud Computing." *Computers, Materials & Continua* 72, no. 2 (2022).

[9] Sultan, Ola Hani Fathi, and Turkan Ahmed Khaleel. "Challenges of Load Balancing Techniques in Cloud Environment: A Review." *Al-Rafidain Engineering Journal (AREJ)* 27, no. 2 (2022): 227-235.

[10] Tawfeeg, Tawfeeg Mohmmed, Adil Yousif, Alzubair Hassan, Samar M. Alqhtani, Rafik Hamza, Mohammed Bakri Bashir, and Awad Ali. "Cloud dynamic load balancing and reactive fault tolerance techniques: a systematic literature review (SLR)." *IEEE Access* 10 (2022): 71853-71873.

[11] Oduwole, Oludayo A., Solomon A. Akinboro, Olusegun G. Lala, Michael A. Fayemiwo, and Stephen O. Olabiyisi. "Cloud Computing Load Balancing Techniques: Retrospect and Recommendations." *J. Eng. Technol* 7, no. 1 (2022): 17-22.

[12] Kaur, Pankaj Deep, and Inderveer Chana. "A resource elasticity framework for QoS-aware execution of cloud applications." *Future Generation Computer Systems* 37 (2014): 14-25.

[13] Abid, Adnan, Muhammad Faraz Manzoor, Muhammad Shoaib Farooq, Uzma Farooq, and Muzammil Hussain. "Challenges and Issues of Resource Allocation Techniques in Cloud Computing." *KSII Transactions on Internet & Information Systems* 14, no. 7 (2020).

[14] Mousavi, Seyedmajid, Amir Mosavi, Annamria R. Varkonyi-Koczy, and Gabor Fazekas. "Dynamic resource allocation in cloud computing." *Acta Polytechnica Hungarica* 14, no. 4 (2017): 83-104.

[15] Devarasetty, Prasad, and Satyananda Reddy. "Genetic algorithm for quality of service based resource allocation in cloud computing." *Evolutionary Intelligence* 14 (2021): 381-387.

[16] Perri, Damiano, Marco Simonetti, Sergio Tasso, Federico Ragni, and Osvaldo Gervasi. "Implementing a scalable and elastic computing environment based on cloud containers." In *Computational Science and Its Applications–ICCSA 2021: 21st International Conference, Cagliari, Italy, September 13–16, 2021, Proceedings, Part I 21*, pp. 676-689. Springer International Publishing, 2021.

[17] Sehgal, Naresh Kumar, Pramod Chandra P. Bhatt, and John M. Acken. "Cloud Computing Scalability." In *Cloud Computing with Security and Scalability. Concepts and Practices*, pp. 241-269. Cham: Springer International Publishing, 2022.

[18] Ngo, Kim Long, Joydeep Mukherjee, Zhen Ming Jiang, and Marin Litoiu. "Evaluating the scalability and elasticity of function as a service platform." In *Proceedings of the 2022 ACM/SPEC on International Conference on Performance Engineering*, pp. 117-124. 2022.

[19] Liu, Xiao-Yang, Zechu Li, Zhuoran Yang, Jiahao Zheng, Zhaoran Wang, Anwar Walid, Jian Guo, and Michael I. Jordan. "ElegantRL-Podracer: Scalable and elastic library for cloud-native deep reinforcement learning." *arXiv preprint arXiv:2112.05923* (2021).

[20] Hussain, Mehboob, Lian-Fu Wei, Abdullah Lakhan, Samad Wali, Soragga Ali, and Abid Hussain. "Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing." *Sustainable Computing: Informatics and Systems* 30 (2021): 100517.

[21] Zeng, Qunsong, Yuqing Du, Kaibin Huang, and Kin K. Leung. "Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing." *IEEE Transactions on Wireless Communications* 20, no. 12 (2021): 7947-7962.

[22] Abdulsalam, Yunusa Simpa, and Mustapha Hedabou. "Security and privacy in cloud computing: technical review." *Future Internet* 14, no. 1 (2021): 11.

[23] Ari, Ado Adamou Abba, Olga Kengni Ngangmo, Chafiq Titouna, Ousmane Thiare, Alidou Mohamadou, and Abdelhak Mourad Gueroui. "Enabling privacy and security in Cloud of Things." (2019).

[24] Gill, Sajid Habib, Mirza Abdur Razzaq, Muneer Ahmad, Fahad M. Almansour, Ikram Ul Haq, N. Z. Jhanjhi, Malik Zaib Alam, and Mehedi Masud. "Security and privacy aspects of cloud computing: a smart campus case study." *Intelligent Automation & Soft Computing* 31, no. 1 (2022): 117-128.

[25] Shen, Jian, Huijie Yang, Pandi Vijayakumar, and Neeraj Kumar. "A privacy-preserving and untraceable group data sharing scheme in cloud computing." *IEEE Transactions on Dependable and Secure Computing* 19, no. 4 (2021): 2198-2210.

[26] Akremi, Aymen, and Mohsen Rouached. "A comprehensive and holistic knowledge model for cloud privacy protection." *The Journal of Supercomputing* (2021): 1-33.

Machines and Algorithms
http://www.knovell.org/mna

Research Article

Automated Detection and Localization of Fungal Infections on Cotton Leaves Using YOLO-based Object Detection Model

Mohammad Sajid Maqbool[1,*], Rubaina Nazeer[1], Abdul Basit[1] and Kinat Zahra[2]

[1]Department of Information Sciences, University of Education, Multan, 60000, Pakistan

[2]Institue of Computer Sciences and Information Technology, The Women University Multan, 60000, Pakistan

[*]Corresponding Author: Muhammad Sajid Maqbool Email: sajidmaqbool7638@gmail.com

Received: 25 May2023; Revised: 15 June 2023; Accepted: 04 August 2023; Published: 16 August 2023

AID: 002-02-000024

Abstract: Cotton is a vital cash crop globally, and its health and productivity are constantly threatened by various diseases. Early detection and accurate diagnosis of these diseases are crucial for effective crop management and minimizing yield losses. In this study, we propose a cotton leaf disease detection system utilizing object detection techniques. Creating an accurate, automated system for spotting and locating illnesses on cotton leaves is the aim of this study. Due to its real-time processing capabilities, we use cutting-edge object detection algorithms, concentrating on the widely used YOLO (You Only Look Once) paradigm. The model is trained using a sizable dataset of cotton leaf photos that have been annotated and creating an xml file and contain samples that have disease infections (fungal). The proposed approach utilizes the ResNet-101 deep convolutional neural network, which has demonstrated strong performance in various computer vision tasks. The model is pretrained on large-scale image datasets to capture high-level features and then fine-tuned on a custom dataset containing annotated cotton leaf images. The dataset used in this research consists of diverse images of cotton plants captured under various environmental conditions. Each image is manually annotated to mark the bounding boxes around individual cotton leaves. These annotations serve as ground truth data for training and evaluating the object detection model. our proposed model achieved an accuracy of 93 percent.

Keywords: Object detection; Cotton Disease Detection; YOLO Model; Cotton Leaf Illness;

1. Introduction

To ensure optimal agricultural output and avoid severe yield losses, crop disease detection is essential. One of the most commercially significant crops in the world, cotton is prone to a number of illnesses that can have a negative influence on both the quality and quantity of the crop. For timely interventions to be put in place and their negative effects on cotton production to be minimized, early diagnosis and correct identification of these illnesses are essential. The "Detection of Disease in Cotton Leaves" project seeks to create an automated system capable of accurately identifying and categorizing illnesses in cotton leaves through visual analysis [2,9,28]. This project aims to create a trustworthy and effective method for farmers and agronomists to detect and control disease outbreaks in cotton crops by utilizing developments in computer vision, machine learning, and image processing techniques. One of the most commercially significant crops in the world, cotton provides the textile sector with essential raw materials. However,

cotton plants are vulnerable to a number of illnesses, which causes substantial productivity losses and financial difficulties for producers. Effective therapy and control of many diseases depend on early detection and precise diagnosis. In recent years, automatic and effective disease identification in plants, particularly illnesses of cotton leaves, has been possible thanks to object detection algorithms. The advantages, difficulties, and possible uses in agricultural practices are highlighted in this overview of cotton leaf disease detection utilizing object detection techniques. Fungal, bacterial, and viral pathogens can cause a number of illnesses that can affect cotton plants [22,30]. Verticillium wilt, Fusarium wilt, Bacterial blight, and Cotton leaf curl virus are among the common diseases that affect cotton leaves. These ailments can cause wilting, defoliation, stunted growth, and other symptoms that have a significant impact on cotton yield. To stop the spread of illness and reduce agricultural losses, prompt identification and management are essential.

The training of Cotton Leaf Disease Detection Model (CLDDM) required large amount of cotton leaf images dataset. Need of implementing quality preprocessing techniques (Resizing of images, Augmentation of images and Normalization of images) to improve the quality of images and convert the dataset into a format that understand by the Object Detection Model. Need of constructing an automated Deep Learning (DL) model that accurately detect and classify the Effected and Healthy Cotton Leaf (CL).

The key objectives of this project are, to train a robust disease detection model, a diverse and representative dataset of cotton leaf images infected with various diseases is collected. These images serve as the foundation for developing an accurate and generalizable disease detection system, the collected dataset undergoes preprocessing techniques to enhance image quality, remove noise, and standardize the data. Augmentation techniques also employed to increase the diversity and variability of the dataset, enabling the model to learn effectively from limited data, State-of-the-art deep learning model, convolutional neural networks (Resnet), employed to build a disease detection model. The model trained on the annotated dataset, learning to recognize and differentiate between healthy and effected cotton leaves, the developed model capable of accurately classifying different affecting cotton leaves, including but not limited to bacterial, viral, or fungal infections. This classification capability to enable farmers to identify specific diseases and take appropriate measures for disease management and treatment.

By implementing an automated disease detection system for cotton leaves, this project aims to empower farmers and agronomists with a valuable tool for early detection and management of diseases. Timely and accurate disease identification can lead to targeted interventions, reducing crop losses, optimizing resource allocation, and ultimately contributing to sustainable and resilient cotton farming practices.

2. Related studies

Many research projects are created for the detection of cotton leaves disease some of them are discussed in this section.

2.1. Related System 1

In [2] an object detection system is created in which they explained that Over 6 million farmers in India depend on cotton as one of their main cash crops, and it is vital to the country's agricultural economy. However, a significant drawback is that the cotton crop is extremely vulnerable to pests and diseases, which results in 30–35% of the harvest being contaminated. As a result, early disease diagnosis is essential because delayed disease detection results in crop failure. Utilizing machine learning and computer vision advancements can therefore be very beneficial to the agriculture industry. In order to identify pests and illnesses on cotton leaves, this research focuses on applying the Mask-RCNN object detection technique, which is based on instance segmentation. Regarding cotton in India's agribusiness, it substantially contributes to the subsistence of about 40–50 million people who work in the agricultural sector. India's horticultural sector is very important to its economy. The handling, trading, and farming of cotton are all important aspects of the material industry and the national economy. India possesses the world's largest cotton-growing area, spanning 126 lakh hectares. Since cotton requires a high temperature of roughly 25 to 30 degrees Celsius, tropical and subtropical regions of the world are the greatest sites to grow it [2].

Cotton is a Kharif crop. Considering the fact that Shirpur, a region in Maharashtra, is home to 24 modern industries. 800 transport fewer weavers produce 1.5 lakh meters of cotton textures daily in Shirpur's Material Park. This group receives its cotton from roughly 3 lakh ranchers and has increased productivity by learning about other things like water harvesting and the use of cash crops. In any case, a number of factors, such as excessive precipitation, temperature variations, inadequate infections, bacterial and parasite diseases, bug attacks, and improper manure application, prevent the growth of the cotton crop. Because they might occur frequently, irritant attacks and infections result in enormous financial losses. The hapless use of synthetics and composts to manage these vermin attacks has led to the development of bug sprays. So, it turns out that early sickness detection can prove beneficial for additional therapy. As a result, we suggest in this study a method for detecting cotton leaf disease employing Mask RCNN and ResNet50 as the architecture's backbone.

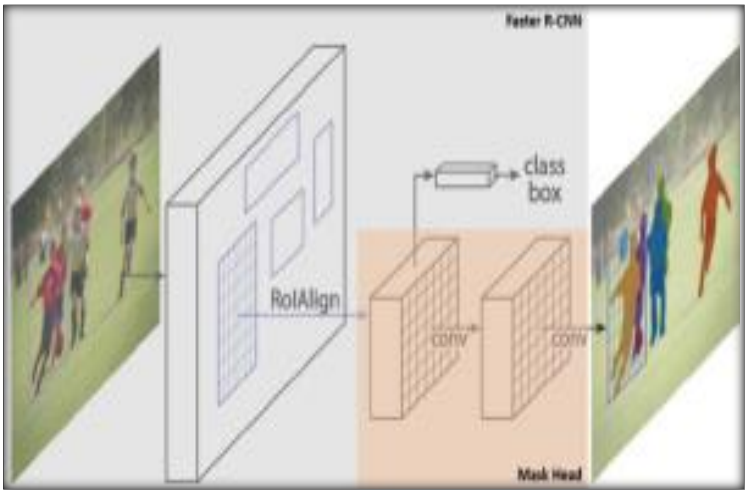

Figure 1: Proposed Architecture of system 1

Contrarily, in a distinct sort of segmentation called Semantic Segmentation, which is used by algorithms like Faster R-CNN, items belonging to the same class cannot be distinguished, making it impossible to forecast where the boundaries would be. Due to this significant drawback of semantic segmentation, Mask RCNN, which is based on instance segmentation, is currently being deployed.

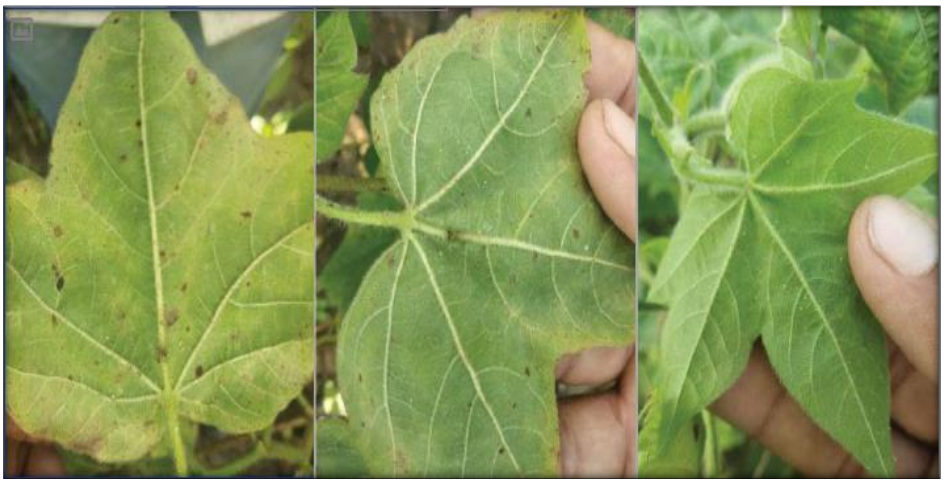

Figure 2: Collected dataset from related system 1

2.2. Related System 2

In related system 2 [3] conduct a study to develop an object detector system for multiclass cotton weed detection. In this work they explain that One of the biggest risks to the production of cotton is weeds. Herbicide resistance in weeds has evolved more quickly as a result of a misuse of pesticides to get rid of weeds, raising worries about the environment, the safety of food, and human health. With the goal of achieving integrated, sustainable weed management, interest in machine vision technologies for artificial or automated weeding is developing. However, the development of trustworthy weed identification and detection technologies continues to be a substantial problem due to the shapeless field environments and important biological heterogeneity of wildflowers. One potential solution to this problem is the development of extensive, labeled pictures of weeds specific to agricultural systems and data-driven artificial intelligence (AI) models for weed detection [21]. Numerous YOLO detectors have garnered significant attention for general object detection and are well-suited for real-time application across various deep learning architectures. In this paper, an additional dataset (CottoWeedDet12) of weed significant to the southern U.S. cotton industry is introduced. It is made up of 9370 bounding box annotations on 5648 photos of 12 distinct weed classes that were taken in cotton fields with natural lighting at different stages of weed growth. A new, extensive A benchmark of 25 state-of-the-art YOLO object detectors of seven versions—YOLO_v3, YOLO_v4, Scaled-YOLO_v4, YOLO_R and YOLO_v5, YOLO_v6, and YOLO_v7—has been built for weed detection on the dataset. YOLOv3-tiny's detection accuracy for mAP@0.5 ranged from 88.14% to 95.22%, whereas Scaled-YOLOv4's accuracy for mAP@ [0.5:0.95] varied from 68.18% to 89.72%. Five replications of Monte-Caro cross validation were used to assess these results. The YOLOv5n and YOLOv5s models in particular have shown a significant deal of promise for cannabis identification in real-time; additionally, data augmentation may increase cannabis detection precision. The weed detection dataset2 and software-programmed algorithms for model benchmarking employed in this study will be useful for future big data and AI-enabled weed detection and control for cotton and possibly other crops.

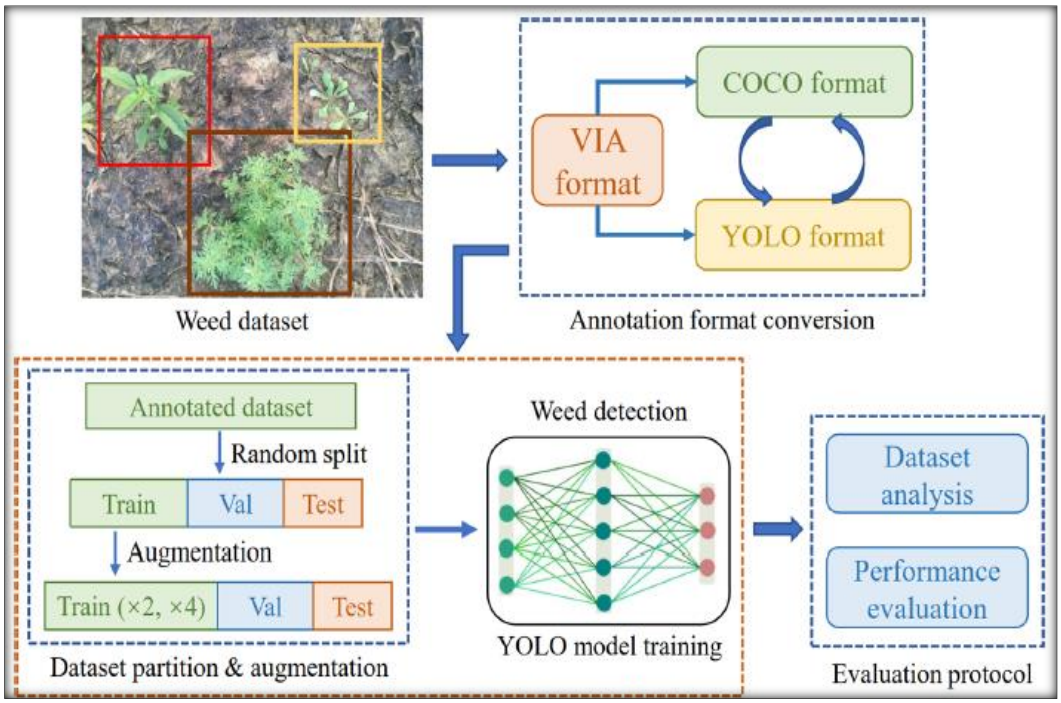

Figure 3: Related System 2

2.3. Other studies

Two important factors that significantly affect the performance of weed recognition are both the amount and quality of the visual data used to train the model and the weed detection techniques. Computer vision algorithms require large-scale labeled picture data to perform successfully. According to study by Sun et al. [4] the performance of advanced deep learning approaches on vision tasks grows logarithmically with the volume of training data. The complete use of deep learning techniques and the creation of reliable machine vision systems in precision agriculture are hindered by the lack of large-scale, high-quality annotated datasets [5]. Good datasets for weed recognition should include appropriate representations of pertinent weed species, environmental factors (such as soil types and light levels), and morphological or physiological changes related to growth stages. In addition to the need for weed detection expertise, creating these datasets is a knowingly costly and time-consuming operation. A number of recent studies, including the Eden Library, Hedge bindweed, CottonWeedID15, Deep Weeds, and Early crop weed dataset, have focused on the creation of image datasets for weed control [6]. To the best of our knowledge, the only available tool for weed identification unique to cotton production systems is CottonWeedID15. However, this dataset is only including image-level annotations, making it unsuitable for applications like weed detection that need bounding box annotations for weed instances in the photographs. While the computations are simple, most of them do not adapt well to changes in imaging settings, particularly when working with images taken under various natural field light situations [7]. CNNs have been applied for weed detection recently, for instance, using data-driven methods based on DL algorithms. Robust against biological variability and imaging circumstances, well-trained deep learning models can reach respectable classification or detection accuracies when fed large-scale datasets [8]. In the interim, a great deal of research has been done on image processing and analysis methods for weed identification [9,15]. For improved weed identification and segmentation from soil backgrounds, a number of color indices that highlight plant greenness have been proposed [10].

Weed identification in plants is a difficult task for ML. On the basis of unmanned helicopters or ground platforms, several automated weed monitor and identification techniques are being developed (Chishun et al., 2019). ML algorithms were paired with handcrafted features that considered a marijuana's, in early weed recognition systems, differences in color, shape, or texture were observed. Support vector machines (SVMs) were employed by the authors to produce local binary features for the classification of agricultural plants. A smaller dataset is frequently required for an SVM's model building. However, it could not be generalizable based on the topic's particulars. DL models are becoming more and more significant in CV because they provide a thorough approach to Identification of weeds for a huge number of datasets that tackles the generalization issues [12]. Sa et al. presented a CNN-based Weednet framework for aerial multispectral photos of sugar beet fields in 2020, and they employed semantic classification for weed detection. Using six experiments, the authors correctly inferred the semantic classes using a cascaded CNN with SegNet applied. The bindweed in the sugar beetroot field dataset was identified by the authors using a YOLO_v3-tiny model. To train the model, they created fake photos and mixed them with actual ones. Using the pooled images, the YOLO_v3 model obtained good detection accuracy. Additionally, weeds can be identified by their trained algorithm in mobile devices and UAVs. The authors of employed an alternative method to recognize weeds in vegetable fields. The authors used the CenterNet model to identify field-grown vegetables before labeling the remaining green spots in the image as weeds [14]. The specific sorts of weeds that exist in fields are ignored by the suggested method, which solely concentrates on crop vegetable identification. In Remote Sens. 2023, 15, 539 4 of 17, the authors presented a thorough examination of the identification of weeds utilizing a 2-stage and a 1-stage detector.

3. Material and Methods

This section describes the project's general research strategy. Indicate the type of approach used, whether experimental, observational, qualitative, quantitative, or a combination of methodologies.

3.1. Architecture of used system

Our used system in dell latitude 6440 and window 10 is installed on it. RAM of 8GB, Hard Disk Drive of 320GB and SSD of 128GB is installed on the used system. Dual core processor is used with two CPU (2.7+2.7). Table 1 list the components of the used system.

Table 1: Used System Specs

Specification	Details
Operating System	Window 10
Used RAM	8GB
Used SSD	128GB
Used HDD	320GB
Software & Tools	Google Colab, MS Word
Language	Python
Model	Dell
Version	Latitude 6440
CPU	2.7+2.7
Generation	4^{th}
Technology	i5
System	Laptop

3.2. Proposed Methodology

Our proposed framework contains two main steps in first step we collect cotton leaf images from different areas of Pakistan such as Multan, Faisalabad etc. The collected images are annotating by using python imglable. The second step of our proposed model is to develop an object detection model to detect cotton disease in the image's dataset.

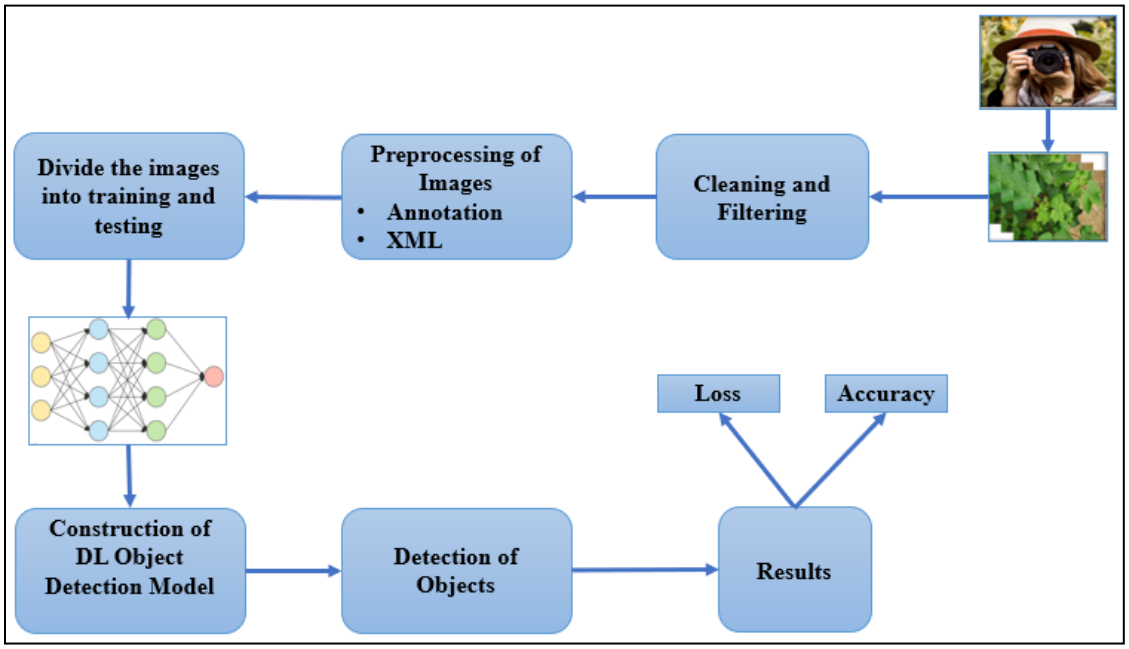

Figure 1: Proposed Framework

Our proposed system firstly focused on collected cotton leaf images dataset and then clean the images by removing blur and unimported images. Secondly preprocessing steps on images are applied to improve the quality of images. thirdly the images are annotating of labeled by using imglable library of python language by creating bonding boxes on the images and create xml file for each image to training the object detection model. Fourthly the annotated images are divided into training and testing images. Finally, an object detection model is developed to detect the disease of the cotton leaf. The steps of proposed methodology are explained below:

3.2.1. Dataset Collection

We collect the dataset from different cities of Pakistan. Four variants from four different areas (ASK1020, MN786, RSK NOOR, and FSD) of cotton leaf images are collected in two classes (Effected and Healthy). There are 501 images are collected. The statics of the images are given in the table 3.2. 130 images are collected from MN786 and 125 images are collected from ASK1020 and 150 images are collected from RSK NOOR and FSD having 96 images.

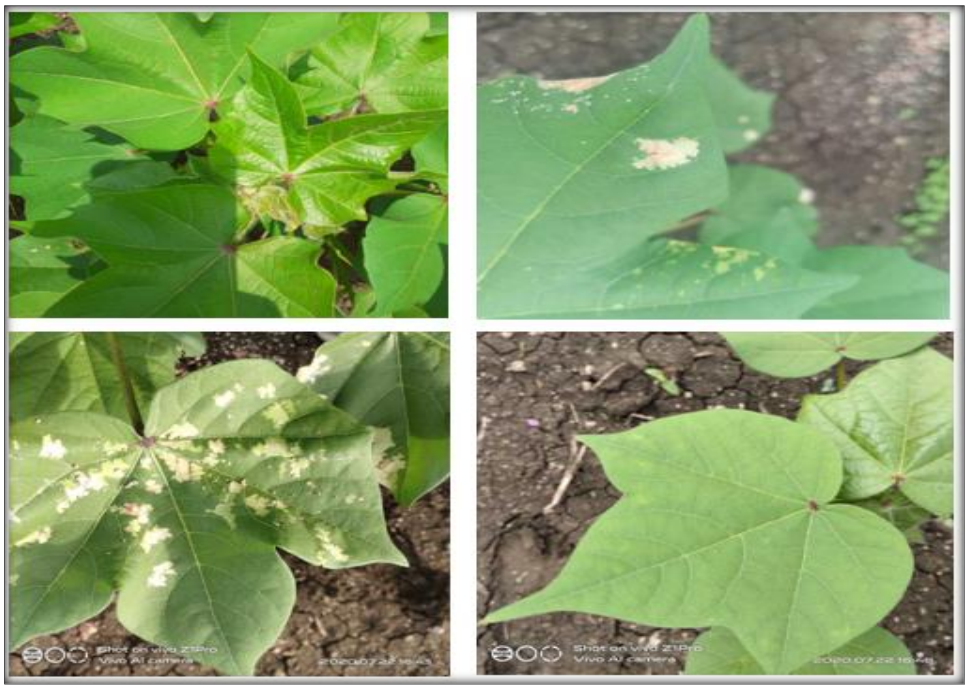

Figure 2: Sample of Collected Images

Table 2: Number of cotton images area wise

Areas	Counting
MN786	130
ASK1020	125
RSK NOOR	150
FSD	96

3.2.2. Cleaning and Filtering

All of the 501 selected images are cleaned and filter. The blur and unimportant images are deleted.

3.2.3. Preprocessing

We perform preprocessing steps on the images to improve the quality of images such as Resizing the images by defining the fix (height and width) of the images and normalization of images.

3.2.4. Annotating

Labelimg library is used to annotate the images. Figure 3.3 shows the sample of one image before annotating and figure 6 shows the image after annotating and creating bonding boxes on the image and create xml file for each image.

Figure 3: Image before Annotation

Figure 4: Image after Annotation

3.2.5. Dividing into training and testing

In this step the dataset is divided into training and testing. eighty percent data is used for the training of object detection model and 20 percent for testing.

Table 3: Dataset Division

Data	No of Images	%
Training	800	80
Testing	201	20

3.2.6. Model creation

We used ssd_resnet101_v1_fpn_640x640_coco17_tpu model. The term "ssd_resnet101_v1_fpn_640x640_coco17_tpu" refers to a specific computer vision model that is used for object detection tasks. Let's break down each component:

- SSD: SSD stands for Single Shot MultiBox Detector. It is a popular object detection algorithm that efficiently detects objects within an image. SSD is known for its real-time processing capabilities.
- ResNet101: ResNet101 is a deep neural network architecture that consists of 101 layers. It is widely used in computer vision tasks due to its ability to effectively learn complex features and patterns from images. v1: This indicates the version of the model. Different versions may have variations in architecture, training techniques, or performance improvements.
- FPN: FPN stands for Feature Pyramid Network. It is a feature extraction technique that enhances the ability of a model to detect objects at different scales. FPN utilizes a top-down and bottom-up pathway to extract features from multiple levels of resolution. 640x640: This indicates the input image size that the model expects. In this case, the model is designed to process images with a resolution of 640x640 pixels.
- COCO17: COCO (Common Objects in Context) is a widely used benchmark dataset for object detection, segmentation, and other related tasks. "COCO17" refers to the 2017 version of the COCO dataset, which contains a large number of labeled images with 80 different object categories.
- TPU: TPU stands for Tensor Processing Unit. It is a specialized hardware accelerator developed by Google for machine learning workloads. TPUs are known for their high-speed and efficient processing, particularly for deep learning tasks. Overall, the "ssd_resnet101_v1_fpn_640x640_coco17_tpu" model combines the SSD algorithm with a ResNet101 backbone, FPN feature extraction, and is trained on the COCO17 dataset. It is designed to perform object detection on images with a resolution of 640x640 pixels using TPU hardware for efficient inference.

3.2.7. ResNet101

Residual Network 101, is a deep CNN architecture that has 101 layers. Microsoft Research first presented it in 2015 as a way to overcome the difficulty of training extremely deep neural networks. The idea of residual learning, which enables the network to learn residual mappings rather than the underlying desired mappings directly, is the fundamental innovation of ResNet101. Introduced "shortcut connections" or skip connections that bypass one or more network levels allow for this to be accomplished. By doing this, the residual information—that is, the difference between the desired output and the current input—can be learned by the network more quickly. The skip connections in ResNet101 enable the network to effectively tackle the problem of vanishing gradients, where the gradients diminish as they propagate backward through the network during training. This issue can make it challenging to train deep networks, as the gradients become too small to effectively update the weights of early layers. ResNet101's skip connections mitigate this issue by permitting the gradients to skip over a number of layers, improving the network's capacity to learn and function.

ResNet101 has been widely adopted and has produced cutting-edge outcomes in a number of computer vision tasks, including object identification, image segmentation, and image classification. Its deep architecture and residual learning concept have proven to be effective in capturing complex features and patterns from images, leading to improved accuracy and generalization. It is worth noting that ResNet101 is just one variant of the ResNet family, which includes different versions with varying depths (e.g., ResNet50, ResNet152). Each variant offers a trade-off between model complexity and performance, allowing practitioners as well as researchers should select the best model according to their needs and available computing power.

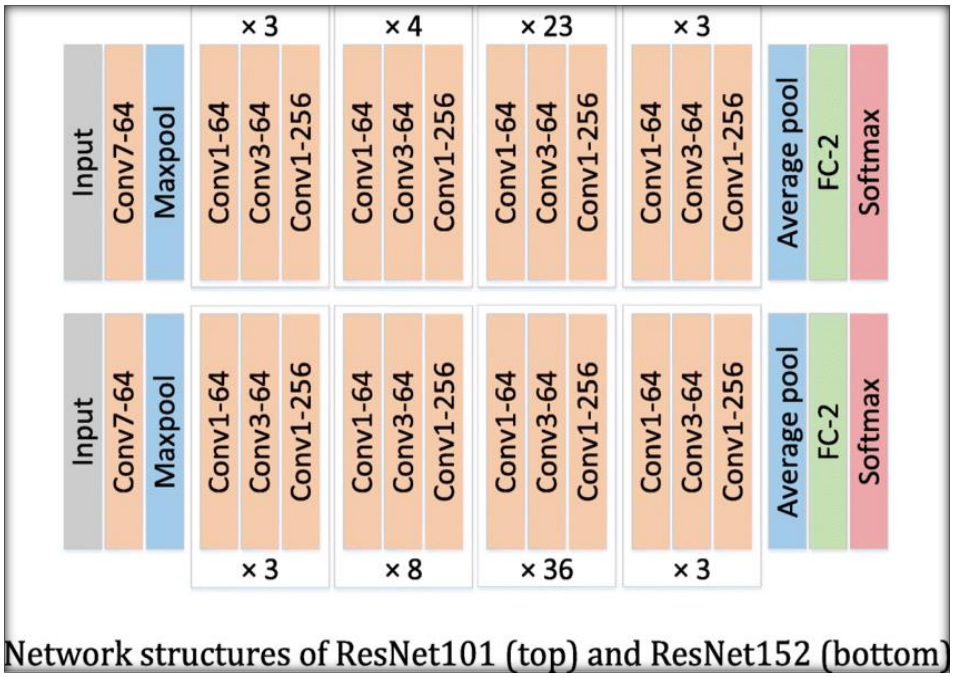

Figure 5: ResNet101 Architecture

4. Experiments and results

In this section of paper, we delve into the exciting realm of experiments and results, where we showcase the empirical evaluation of the proposed model. Here, we present a comprehensive analysis of the performance and efficacy of the system in various scenarios and benchmarks. Through rigorous experimentation, we aim to provide insights into the capabilities, limitations, and potential applications of the model. This chapter's major goal is to evaluate the model's performance and applicability for the tasks at hand. We address crucial issues like: Can the model successfully identify items across a range of images? Which scales, orientations, and occlusions does it handle best? What effect do differ input resolutions have on the speed and accuracy of detection? Through methodical experiments and careful evaluation, these issues and others are investigated. We make use of well-known datasets like COCO17, which offers a wide range of images annotated with object descriptions, to carry out the tests. Advanced methods and architectures, like the SSD (Single Shot MultiBox Detector), ResNet101 backbone, and FPN (Feature Pyramid Network), are used to train the model. To improve the model's capability to recognized things reliably and effectively, these elements are carefully mixed.

4.1. Experimental setup

The experimental setup of our proposed model is given below:

Table 4: Model Architecture

Parameters	Detail
Classes	2
Number of epochs	10
Batch size	16
Loops	2000
Depth	256
Images size	640*640
num_layers_before_predictor:	4
Kernal_Size	3

The above parameters are chosen because result was good on these parameters. We evaluate our proposed model on different parameters other than mentioned above but result was not good.

4.2. Results

We evaluate our model by testing 5 different images and record its results:

4.2.1. Image test 1

We test an image on the trained Object Detection Model (ODM). Table 5 shows the result of first tested image.

Table 5: Results of Test image 1

Number of Turns	Accuracy	Learning Rate (%)	Lose	Average Loss
200	76	7	76	60
400	77	12	34	36
600	75	15	40	35
800	80	15	35	32
1000	85	16	33	28
12001	86	23	32	25
1400	88	26	25	19
1600	92	39	19	15
1800	93	56	11	10
2000	93	60	6	3

Table 5 shows that model gives accuracy of 76, 77, 75, 80, 85, 86, 88, 92, 93 and 93 percent on the turns (200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800 and 2000). Learning rate of 7, 12, 15, 15, 16, 23, 26, 39, 56 and 60 percent is achieved by the model on (200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800 and 2000) turns on image 1. Model on Image 1 gives loss of 76, 34, 40, 35, 33, 32, 25, 19, 11 and 6 on the 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800 and 2000 turns respectively. Average loss of model on image one is 60, 36, 35, 32, 28, 25, 19, 15, 10, 3 on (200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800 and 2000) turns respectively.

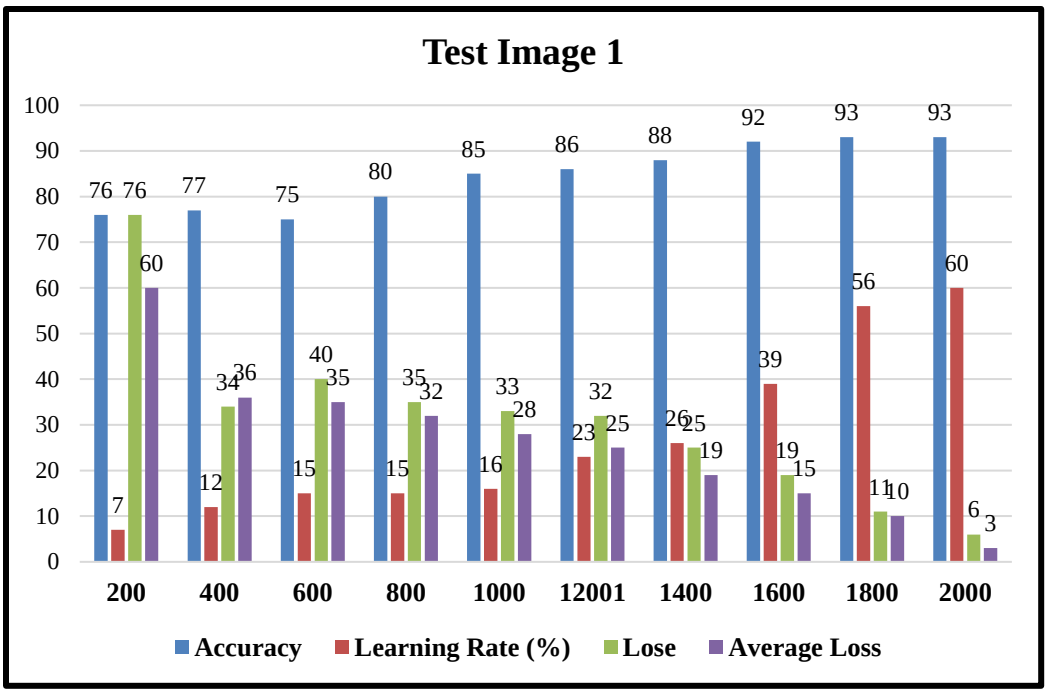

Figure 6: Result of Test Image 1

4.2.2. Images test 2

We test an image on the trained Object Detection Model (ODM). Table 6 shows the result of Second tested image.

Table 6: Results of Test image 2

Number of Turns	Accuracy	Learning Rate (%)	Lose	Average Loss
200	60	9	80	56
400	65	11	76	57
600	67	14	65	51
800	68	14	59	45
1000	70	18	46	42
1200	78	21	41	34
1400	79	23	39	30
1600	84	31	20	24
1800	86	45	15	20
2000	90	55	10	13

Table 6 shows that model gives accuracy of 60,65, 67, 68, 70, 78, 79, 84, 86 and 90 percent on the turns (200, 400, 600, 800, 1000, 1200, 1400, 1600, 18020, 15, 100 and 2000). Learning rate of 9, 11, 14, 14, 18, 21, 23, 31, 45, and 55 percent is achieved by the model on (200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800 and 2000) turns on image 1. Model on Image 1 gives loss of 80, 76, 65, 59, 46, 41, 39,20, 15, and 10 on the 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800 and 2000 turns respectively.

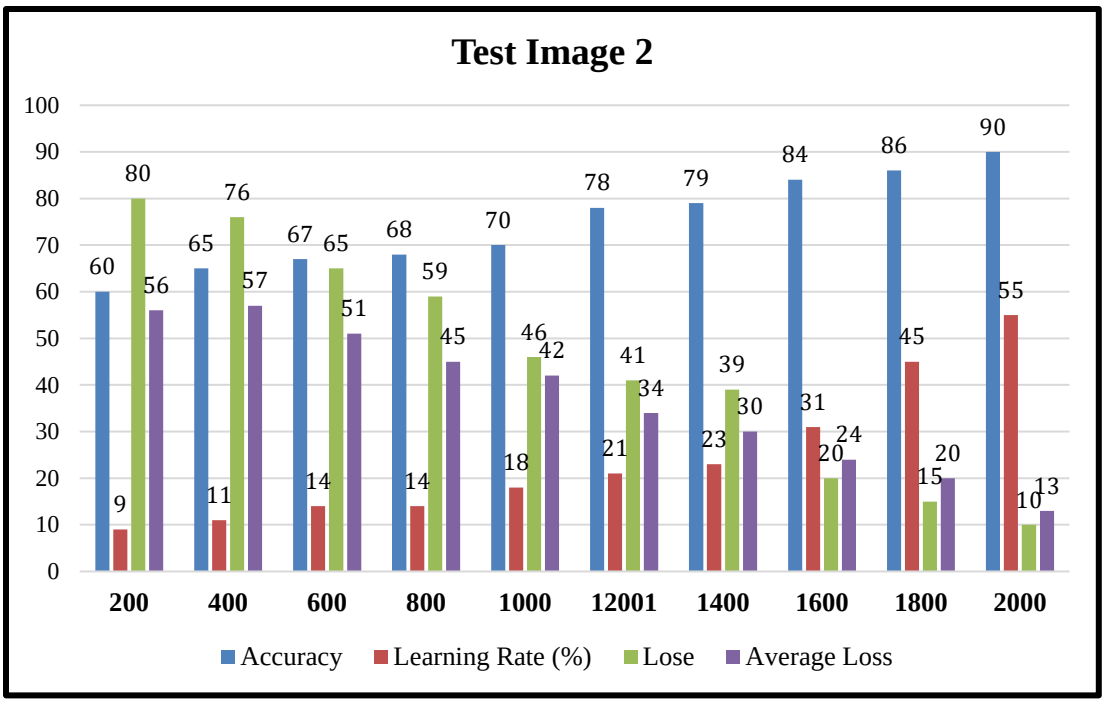

Figure 7: Results of Test Image 2

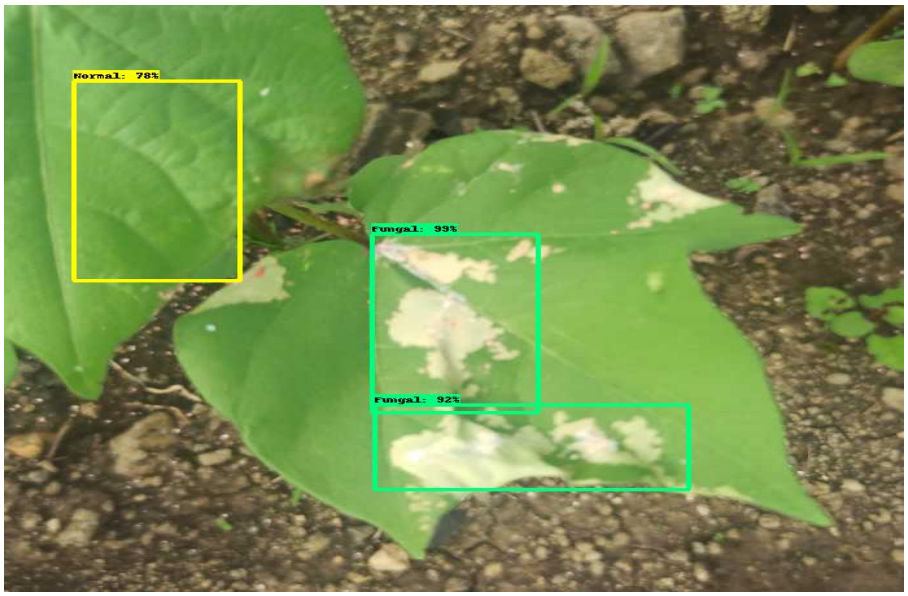

Figure 8: Proposed Model Tested Image Screenshot 1

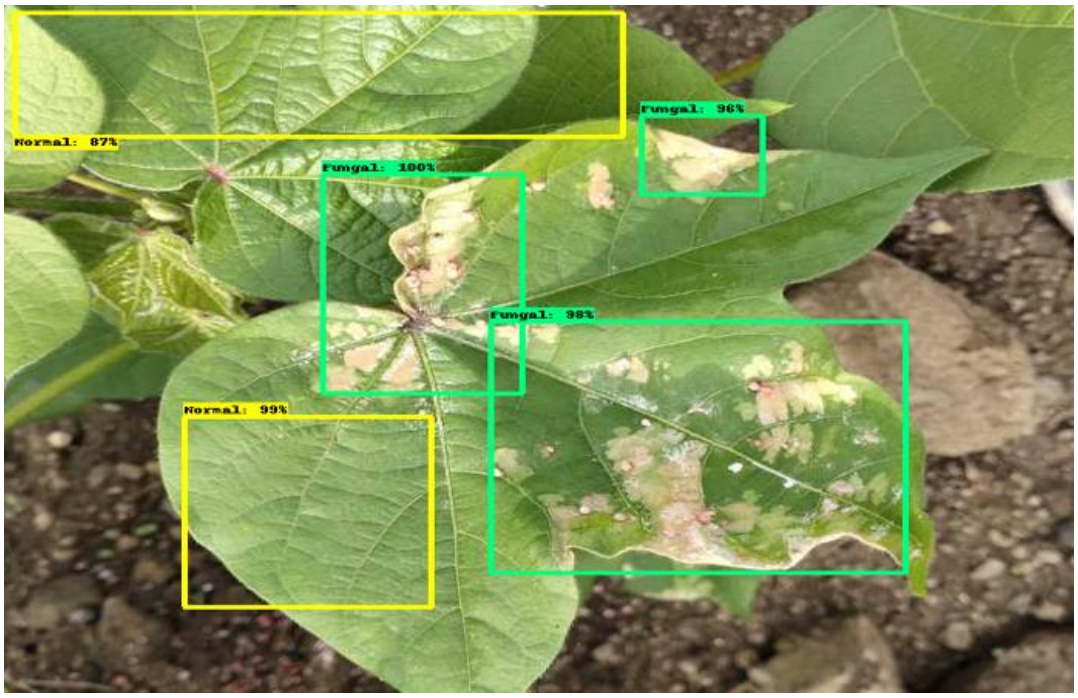

Figure 9: Proposed Model Tested Image Screenshot 2

The results of our proposed model are best as compared to the literatures studies. Proposed experiment results are 10 percent higher than the above-mentioned studies.

5. Conclusion

In this study, we have developed an automated cotton leaf disease detection system using object detection techniques. The proposed approach combines the YOLO paradigm with the ResNet-101 deep convolutional neural network to accurately identify and locate diseases on cotton leaves. The system achieved an impressive accuracy of 93 percent with a low error rate of 6 percent.

We were able to train and fine-tune the model successfully thanks to the use of a sizable and varied dataset and hand annotation of bounding boxes. The ResNet-101 model was able to capture high-level features important for cotton leaf disease diagnosis since it had been pretrained on large picture datasets. The outcomes show the system's potential to help farmers and agricultural specialists identify and diagnose illnesses on cotton leaves early on. The spread of infections can be controlled and yield losses can be kept to a minimum by rapidly detecting unhealthy plants and implementing the necessary remedies.

References

[1] Remya, S., T. Anjali, S. Abhishek, Somula Ramasubbareddy, and Yongyun Cho. "The Power of Vision Transformers and Acoustic Sensors for Cotton Pest Detection." *IEEE Open Journal of the Computer Society* (2024).

[2] Dang, Fengying, Dong Chen, Yuzhen Lu, and Zhaojian Li. "YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems." *Computers and Electronics in Agriculture* 205 (2023): 107655.

[3] Huang, Xin, Xinxin Wang, Wenyu Lv, Xiaying Bai, Xiang Long, Kaipeng Deng, Qingqing Dang et al. "PP-YOLOv2: A practical object detector." *arXiv preprint arXiv:2104.10419* (2021).

[4] arXiv preprint arXiv:2104.10419. Jocher, G., 2020. yolov5. Code repository. *https://github.com/ultralytics/yolov5*.

[5] Kennedy, HannahJoy, Steven A. Fennimore, David C. Slaughter, Thuy T. Nguyen, Vivian L. Vuong, Rekha Raja, and Richard F. Smith. "Crop signal markers facilitate crop detection and weed removal from lettuce and tomato by an intelligent cultivator." *Weed Technology* 34, no. 3 (2020): 342-350.

[6] Kniss, Andrew R. "Genetically engineered herbicide-resistant crops and herbicide-resistant weed evolution in the United States." *Weed Science* 66, no. 2 (2018): 260-273.

[7] Lamm, Ross D., David C. Slaughter, and D. Ken Giles. "Precision weed control system for cotton." *Transactions of the ASAE* 45, no. 1 (2002): 231.

[8] Lati, Ran Nisim, Jesper Rasmussen, Dionisio Andujar, Jose Dorado, Therese W. Berge, Christina Wellhausen, Michael Pflanz et al. "Site-specific weed management—constraints and opportunities for the weed research community: Insights from a workshop." *Weed Research* 61, no. 3 (2021): 147-153.

[9] Legleiter, Travis R., and Kevin W. Bradley. "Glyphosate and multiple herbicide resistance in common waterhemp (Amaranthus rudis) populations from Missouri." *Weed Science* 56, no. 4 (2008): 582-587.

[10] Li, Chuyi, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke et al. "YOLOv6: A single-stage object detection framework for industrial applications." *arXiv preprint arXiv:2209.02976* (2022).

[11] Li, Yong, Zhiqiang Guo, Feng Shuang, Man Zhang, and Xiuhua Li. "Key technologies of machine vision for weeding robots: A review and benchmark." *Computers and Electronics in Agriculture* 196 (2022): 106880.

[12] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740-755. Springer International Publishing, 2014.

[13] Liu, Li, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. "Deep learning for generic object detection: A survey." *International journal of computer vision* 128 (2020): 261-318.

[14] Lu, Y. "CottonWeedDet12: a 12-class weed dataset of cotton production systems for benchmarking AI models for weed detection [Data set]. Zenodo." (2023).

[15] Lu, Yuzhen, Dong Chen, Ebenezer Olaniyi, and Yanbo Huang. "Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review." *Computers and Electronics in Agriculture* 200 (2022): 107208.

[16] Lu, Yuzhen, and Sierra Young. "A survey of public datasets for computer vision tasks in precision agriculture." *Computers and Electronics in Agriculture* 178 (2020): 105760.

[17] Lu, Yuzhen, Sierra Young, Haifeng Wang, and Nuwan Wijewardane. "Robust plant segmentation of color images based on image contrast optimization." *Computers and Electronics in Agriculture* 193 (2022): 106711.

[18] MacRae, A. W., T. M. Webster, L. M. Sosnoskie, A. S. Culpepper, and J. M. Kichler. "Cotton yield loss potential in response to length of Palmer amaranth (Amaranthus palmeri) interference." *J Cotton Sci* 17, no. 3 (2013): 227-32.

[19] Manalil, Sudheesh, Onoriode Coast, Jeff Werth, and Bhagirath Singh Chauhan. "Weed management in cotton (Gossypium hirsutum L.) through weed-crop competition: A review." *Crop Protection* 95 (2017): 53-59.

[20] Meyer, George E., and Joao Camargo Neto. "Verification of color vegetation indices for automated crop imaging applications." *Computers and electronics in agriculture* 63, no. 2 (2008): 282-293.

[21] Misra, Diganta. "Mish: A self regularized non-monotonic activation function." *arXiv preprint arXiv:1908.08681* (2019).

[22] Morgan, Gaylon D., Paul A. Baumann, and James M. Chandler. "Competitive impact of Palmer amaranth (Amaranthus palmeri) on cotton (Gossypium hirsutum) development and yield." *Weed Technology* 15, no. 3 (2001): 408-412.

[23] Mylonas, Nikos, Ioannis Malounas, Sofia Mouseti, Eleanna Vali, Borja Espejo-Garcia, and Spyros Fountas. "Eden library: A long-term database for storing agricultural multi-sensor datasets from uav and proximal platforms." *Smart Agricultural Technology* 2 (2022): 100028.

[24] Smart Agric. Technol. 2, 100028 https://doi.org/10.1016/j. atech.2021.100028. Nelson, J., Solawetz, J. (2020). Responding to the controversy about yolov5. *https://blo g.roboflow.com/yolov4-versus-yolov5/*,

[25] Nepal, Upesh, and Hossein Eslamiat. "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs." *Sensors* 22, no. 2 (2022): 464.

[26] Olsen, Alex, Dmitry A. Konovalov, Bronson Philippa, Peter Ridd, Jake C. Wood, Jamie Johns, Wesley Banks et al. "DeepWeeds: A multiclass weed species image dataset for deep learning." *Scientific reports* 9, no. 1 (2019): 2058.

[27] Maqbool, Muhammad Sajid, Israr Hanif, Sajid Iqbal, Abdul Basit, and Aiman Shabbir. "Optimized Feature Extraction and Cross-Lingual Text Reuse Detection using Ensemble Machine Learning Models." *Journal of Computing & Biomedical Informatics* 5, no. 01 (2023): 26-40.

[28] Malik, Fiza, Muhammad Fuzail, Naeem Aslam, Ramla Sarwar, Kamran Abid, Muhammad Sajid Maqbool, and Anum Yousaf. "A Hybrid Machine Learning Model to Predict Sentiment Analysis on X." *Journal of Computing & Biomedical Informatics* 6, no. 02 (2024): 64-79.

[29] Hasnain, Muhammad Adnan, Sadaqat Ali, Hassaan Malik, Muhammad Irfan, and Muhammad Sajid Maqbool. "Deep learning-based classification of dental disease using X-rays." *Journal of Computing & Biomedical Informatics* 5, no. 01 (2023): 82-95.

[30] Fazal, Unaiza, Muhibullah Khan, Muhammad Sajid Maqbool, Hadia Bibi, and Rubaina Nazeer. "Sentiment Analysis of Omicron Tweets by using Machine Learning Models." *VFAST Transactions on Software Engineering* 11, no. 1 (2023): 67-75.

[31] Kanwal, Fouzia, Mr Kamran Abid, Muhammad Sajid Maqbool, Naeem Aslam, and Muhammad Fuzail. "Optimized Classification of Cardiovascular Disease Using Machine Learning Paradigms." *VFAST Transactions on Software Engineering* 11, no. 2 (2023): 140-148.

[32] Rafiqee, Muhammad Mussadiq, Zahid Hussain Qaiser, Muhammad Fuzail, Naeem Aslam, and Muhammad Sajid Maqbool. "Implementation of Efficient Deep Fake Detection Technique on Videos Dataset Using Deep Learning Method." *Journal of Computing & Biomedical Informatics* 5, no. 01 (2023): 345-357.

[33] Abid, Kamran, Naeem Aslam, Muhammad Fuzail, Muhammad Sajid Maqbool, and Kainat Sajid. "An Efficient Deep Learning Approach for Prediction of Student Performance Using Neural Network." *VFAST Transactions on Software Engineering* 11, no. 4 (2023): 67-79.

[34] Padilla, Rafael, Sergio L. Netto, and Eduardo AB Da Silva. "A survey on performance metrics for object-detection algorithms." In *2020 international conference on systems, signals and image processing (IWSSIP)*, pp. 237-242. IEEE, 2020.

[35] Pandey, Piyush, Hemanth Narayan Dakshinamurthy, and Sierra N. Young. "Frontier: autonomy in detection, actuation, and planning for robotic weeding systems." *Transactions of the ASABE* 64, no. 2 (2021): 557.

[36] Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen et al. "Pytorch: An imperative style, high-performance deep learning library." *Advances in neural information processing systems* 32 (2019).

[37] Peruzzi, Andrea, Luisa Martelloni, Christian Frasconi, Marco Fontanelli, Michel Pirchio, and Michele Raffaelli. "Machines for non-chemical intra-row weed control in narrow and wide-row crops: a review." *Journal of Agricultural Engineering* 48, no. 2 (2017): 57-70.

[38] Raschka, Sebastian. "Model evaluation, model selection, and algorithm selection in machine learning." *arXiv preprint arXiv:1811.12808* (2018).

Machines and Algorithms
http://www.knovell.org/mna

Research Article

Clustering Algorithms: An Investigation of K-mean and DBSCAN on Different Datasets

Arooj Zahra[1] and Nabeel Asghar[1]

[1]Department of Computer Science, Bahuddin Zakariya University, Multan, 60000, Pakistan

[*]Corresponding Author: Arooj Zahra. Email: aroojzahra97@gmail.com

Received: 06 June 2023; Revised: 21 June 2023; Accepted: 08 August 2023; Published: 16 August 2023

AID: 002-02-000025

Abstract: The branch of artificial intelligence that studies computer techniques that allow systems to learn autonomously and deliver outcomes based on past experience without being programmed. Supervised and unsupervised machine learning are major categories. Our research focuses on unsupervised learning with unlabeled data. Clustering is an unsupervised learning method that groups unlabeled data items by similarity. Several studies have compared clustering algorithms based on complexity, performance, and the impact of cluster number on performance. To our knowledge, no study has evaluated clustering methods on small and large datasets. A detailed study was conducted to evaluate DB-SCAN and K-Means algorithms on small and large datasets. We have collected 17 open access, publicly available machine learning heterogeneous datasets from online machine learning dataset sources such as the UCI repository, Keel, and Kaggle. The datasets are divided into small and large categories based on the number of instances in each dataset. Different preprocessing techniques are used to improve the quality of datasets. The class field is removed from the preprocessed datasets and then put into the two clustering techniques outlined above. The clustered data is analyzed using three classifiers (K-Nearest Neighbor, Support Vector Machine, and Naïve Bayes) to evaluate the clustering algorithm's performance. The accuracy of the KNN, SVM, and NB classifiers was calculated as part of the final algorithm performance study. The final analysis of tests found that the K-Means algorithm performs better on large datasets, whereas the DB-SCAN clustering technique is more efficient on small datasets.

Keywords: Unsupervised machine learning; Clustering algorithms; DB-SCAN; K-Means; Classifiers;

1. Introduction

In our study, we have compared the performance of two different clustering algorithms, i.e., K-Means and DB-SCAN, over large and small datasets. Data mining refers to the process of extracting useful information from massive datasets by examining records stored in various types of repositories, databases, or data warehouses. Information management, query processing, and decision-making are all possible with this mined data. We employ both supervised and unsupervised machine learning approaches for this goal, with clustering being a common unsupervised method. Clustering organizes datasets into groups of items that share high levels of similarity. The five most commonly used clustering algorithms are as follows: Some examples of these strategies include 1) hierarchical, 2) partitioning, 3) density-based, 4) model-based,

and 5) grid-based. These groups contain algorithms that have undergone testing and evaluation based on criteria such as complexity, speed, scalability, and efficiency. Multiple analyses have contrasted various clustering methods. To illustrate the point, [1] compared the performance of EM and K-means clustering algorithms using a single dataset. In terms of accuracy, performance, and quality, another study [2] compared hierarchical clustering algorithms with soft clustering techniques like K-Means and EM. Additional research [2] compared hierarchical-based methods to partition-based ones based on dataset size, kind, and cluster count. In [3], researchers demonstrated that the CirCle method outperformed other clustering methods based on models or partitions. A comprehensive study [4] compared techniques based on density, hierarchies, grids, as well as different cluster sizes and nested cluster structures. Other research [5] has tested several clustering techniques using various criteria. There is a dearth of research on the optimal clustering methods for big and small datasets, despite the abundance of studies devoted to determining the optimal clustering algorithm in terms of performance efficiency, training time, and complexity. The goal of this research is to determine which clustering algorithms work best with large-scale datasets and which ones work best with small datasets.

In state-of-the-art comparative studies of clustering algorithms' performance, researchers have only focused on the impact of number of clusters on efficiency or compared hierarchal and partition-based clustering techniques. To our knowledge, no study has examined how dataset size affects clustering performance. Any machine learning algorithm's performance depends on dataset size. Small datasets may cause the ML model to overfit or not learn patterns. Thus, dataset size matters when comparing attributes of dataset instances or objects. However, large-scale high-quality datasets are rare, therefore researchers usually have to work with smaller datasets. Researchers benefit from finding a good ML algorithm in such instances. Our study examined two data clustering methods in light of this. This study will compare the performance of the DBSCAN and K-Means clustering methods on datasets of varied sizes. The initial stage will be to locate suitable datasets for the investigation. The datasets will then be separated into two categories: small-scale and large-scale. Finally, the core analysis will compare the performance of DBSCAN and K-Means on the categorized datasets.

In this section, we have already discussed the rationale for our research, the precise problem statement, and the primary aims and objectives. We have conducted a comprehensive literature review in the subsequent section. The methodology of our investigation is thoroughly examined in section 3. The implementation of our proposed adaptive model and the results analysis are discussed in Section 4. In Section 5, the conclusion and the intended future work of our study are discussed.

2. Literature Survey

In this article [8], a hybrid krill herd has been proposed, which includes a harmony search algorithm with a new probability value to regulate the harmony search operator during the exploration search. They have implemented the following metrics to assess their proposed methodology: precision, recall, accuracy, ASDC, and F-measure. Additionally, they have implemented the Error rate and objective function for data clustering. Their results have demonstrated that their algorithm is highly effective. In this paper [9], their objective was to suggest an optimal solution for network communications in a disaster situation that does not involve any disconnectivity, including functional and non-functional areas. They have successfully accomplished their objective; however, their proposed approach is insufficient to function independently. In order to improve the post-disaster situation and establish a stronger connection, they must incorporate additional restoration and protection techniques. This article conducted a survey [10] that employed four clustering methods: LVQ, SOM, COBWEB, and k-means. K-means clustering was the most effective algorithm in their experimentation, as it required less computational effort. The WEKA tool was employed for the experimentation, so it is uncertain whether k-means will perform as well on other tools.

A comparison of supervised and unsupervised machine learning algorithms was conducted [6] using a lung cancer dataset. The combination of Apriori and k-means algorithms results in a quicker performance, as demonstrated by their experimental results and comparison. They have introduced a novel multi-hop clustering algorithm in this article [8]. Cluster head selection mechanism for optimal cluster head selection

is a cluster model that may be presented in their scheme, which is based on priority neighbor-based technique. From their experiments, they have proposed an algorithm that enhances the reliability and stability of VANET. Jin Wang [11] has introduced the particle swarm optimization, a clustering algorithm that includes mobile sink support for WSNs. Their proposed scheme outperformed the three conventional routing algorithms for WSNs that were previously in use, as indicated by the results of their evaluation. In order to enhance the network's efficacy, they implemented the PSO algorithm and the virtual clustering technique.

The Bird Flock Gravitation Searching Algorithm (BFGSA) is a clustering algorithm that has been proposed in this article [12]. The BFGSA is implemented to monitor the progression of candidate clustering centroids in order to identify the robust data cluster in a multi-dimensional Euclidean space. The rate of error and the sum of intra-clustering distance are used to evaluate the performance using thirteen distinct datasets. The performances of k-means, PSO, and GSA are compared. The experimental results indicate that it is straightforward to implement data clusters. This article [13] proposes a routine base protocol known as LEACH-SF, which is an energy-efficient clustering. The fuzzy c-means clustering algorithm was employed to accomplish the balanced clusters. The simulation results indicate that they are capable of constructing efficient balanced clusters and maximizing the network lifetime. LEACH-SF outperformed the classic and fuzzy-clustering algorithms, which are employed to optimize the number of data packets received and minimize the intra-cluster distance. The high-dimension data was proposed to be selected by a sub-set feature clustering-base feature in this paper [7]. Clusters are features in the proposed algorithm, and each cluster is considered as a single feature. A comparison has been conducted with the renowned feature selection algorithms, including FCBF, Relief, CFS, INTERACT Consisting, and FOCUS. This algorithm has achieved the highest proportion of pre-selected features, more precise results, and a shorter runtime for RIPPER, Naïve Bayes, and C4.5, and the second-best efficacy for IB1.

This paper [12] proposes a hybrid PSO algorithm with GOs (H-FSPSOTC) for the selection of text features. The text features selected in the selection method are subsequently utilized by k-means to generate more precise clusters. The H-FSPOSTC results were the most favorable among the other comparatives. This algorithm will assist in the development of enhanced text features and text clustering techniques, such as k-means. Another algorithm has been proposed for ad-hoc networks that is based on grey wolf clustering in another study [14]. Optimize the number of clusters that have been derived from the convergence of the value of α wolf in order to achieve superior results. The simulation is conducted using MATLAB, and the results are compared to those of PSO, CLPSO, and MOPSO. The performance of the proposed method was superior to that of CLIPSO and MOPSO in terms of the number of clusters with varying transmissions, the number of nodes, and the size of the grid. Additionally, it reduces the necessary number of clusters to reduce the cost of routing for communication. This article [15] introduces a paradigm of multi-hop sensor networks known as Type 2-Fuzzy logic. The results of their simulation indicate that T2FL is more scalable, reliable, and superior to the T1FL, LEACH single hop, and LEACH multi-hop protocols.

RNN-DBSCAN, a cutting-edge clustering technique based on density, has been proposed by merging the idea of observation reachability definitions with the observation of reverse nearest neighbor. [16]. According to the evaluation results, the RNN-DBSCAN outperforms the DBSCAN; yet, it is a sophisticated method that can be improved.

Another work described an unsupervised machine learning algorithm whose main goal is to learn a finite mixture model using multivariate data. The term "unsupervised" refers to two properties of the algorithm: 1) it has the ability to choose the number of components, and 2) it must be carefully started, similar to the classic expectation-maximization (EM) technique.[17] Another disadvantage of EM mixture fitting that has been addressed by the presented method is the possibility of the algorithm converging to a unique estimate at the parameter space boundary. The suggested model is unique in that it does not require any model selection criteria. The provided approach integrates both the model selection and estimation processes. The proposed technique can be applied to any parametric mixture model for which an EM algorithm has been created. This fact was demonstrated in this work through experiments, specifically the use of Gaussian mixtures. All of the experiments in this study are designed to test the efficacy of the provided approach.

A comparison was conducted between supervised and unsupervised machine learning techniques [18] using the lung cancer dataset. The experimental results and comparison indicate that the combination of apriority and k-means algorithms results in a more efficient performance. This article conducted a survey [19] that employed four clustering methods: LVQ, SOM, COBWEB, and k-means. K-means clustering was the most effective algorithm in their experimentation, as it required less computational effort. The WEKA tool was employed for the experimentation, so it is uncertain whether k-means will perform as well on other tools. In another research study, the author has addressed the scenarios in which it is highly important to minimize the error of generalization, such as in the case of achieving excellent classification results, or in the case of the occurrence of little bit model over-fitting, which results in a critical penalty in the testing data results. In order to address these circumstances, the application of a classifier with minimal dimensions in Vapnik-Chervonenkis (VC) could result in positive distinctions. This is due to the presence of two benefits: 1) the classifier's learning power is effective even on a small number of instances, and 2) the classifier has the ability to maintain the distance between the training and testing errors. The author of this study has experimentally demonstrated that the application of a classifier known as the majority vote point (MVP) on the basis of a limited number of dimensions in VS can accomplish a lower error of generalization than any other linear classifier. A maximum bound has been established for the dimensions of the VC in the MVP classifier. In the subsequent phases, empirical analysis is employed to predict the precise dimensions of VC. The proposed method is subsequently revalidated by its application to the diagnosis of prostate cancer and the detection of machine defects, which demonstrates that the MVP classifier can achieve a significantly lower generalization error [20].

In an additional investigation [21], the computation of the distance measure for each object in the dataset dominates the computational complexity of DBSCAN. The efficiency of DBSCAN can be enhanced by reducing the complexity of nearest neighbor search for each region query and by reducing the number of region queries (DBSCAN variant). This study conducted a comparative evaluation of the efficacy and effectiveness of clustering in these region queries for DBSCAN. The study concluded that the DBSCAN variant is slightly less effective than DBSCAN, but it significantly enhances efficiency.

In an additional research study [22], the author enhanced the algorithm's global search capability and introduced a semi-supervised K clustering algorithm. Initially, the K-means clustering algorithm was implemented to manage gene data. Then, the greedy iteration was employed to identify the K mean clustering in order to obtain superior results, utilizing the enhanced semi-supervised K mean clustering. The results of the simulation experiment demonstrated that the global semi-supervised K clustering algorithm has a superior cluster effect and optimization capability in comparison to the MDO algorithm. In this context [23], the researcher conducted a systematic comparison of nine well-known clustering methods that are available in the R language, assuming that the data is normally distributed. The researcher considered artificial datasets with a variety of tunable properties, such as the number of classes and the separation between classes, in order to account for the numerous potential variations of data. The assessment of the clustering methods' sensitivity to the various parameters that have been configured. The conclusion demonstrated that the spectral approach exhibited exceptional performance when the default configurations of the adopted methods were taken into account. Additionally, they discovered that the default configuration of the implemented implementations was not consistently precise. In these instances, a straightforward method that relies on the random selection of parameter values was found to be an effective alternative for enhancing performance.

In an additional article [24], the outlier of customer data was identified in order to ascertain customer behavior. The RFM (Recency, Frequency, and Monetary) models were used to determine the customer behavior by clustering the customer data using the K-Mean and DBSCAN algorithms. The investigation has concluded that the outlier in cluster 1 had a 100% similarity in DBSCAN and K-Means. However, the aggregate similarity of the outlier is 67%. The behavior of customers was characterized by a high monetary value but a low frequency of expenditure, as evidenced by the outliers. In this study [46], the researcher proposed a novel K-Means clustering algorithm to resolve the issue of a higher probability of combining dissimilar items into the same group when the number of clusters is limited. Dynamic data clustering was

implemented by the proposed methodology. Initially, the threshold value was determined as the centroid of K-Means in the proposed method, and the number of clusters was generated from this value. A pair of data points is considered to be in the same group if the Euclidian distance between them is less than or equal to the threshold value at each iteration of K-Means. Otherwise, the proposed method will generate a new cluster that contains the dissimilar data point. It has been demonstrated that the proposed approach outperforms the original K-Means method. Clustering and other statistical tools and methods were employed to evaluate students' performance in an additional research study [26]. In this investigation, the K-mean clustering algorithm was implemented. The elbow method was employed to determine the appropriate number of clusters. The analysis was conducted on a gender basis to determine whether there was a pattern based on the gender of the students. The study's findings were that the data was clustered such that data within the same cluster were similar, while data within separate clusters were not.

3. Proposed Methodology

This section describes our research technique, which is to compare K-Means and DB-SCAN clustering algorithms on small and large datasets. Figure 1 below shows our research-adapted model for this assignment.

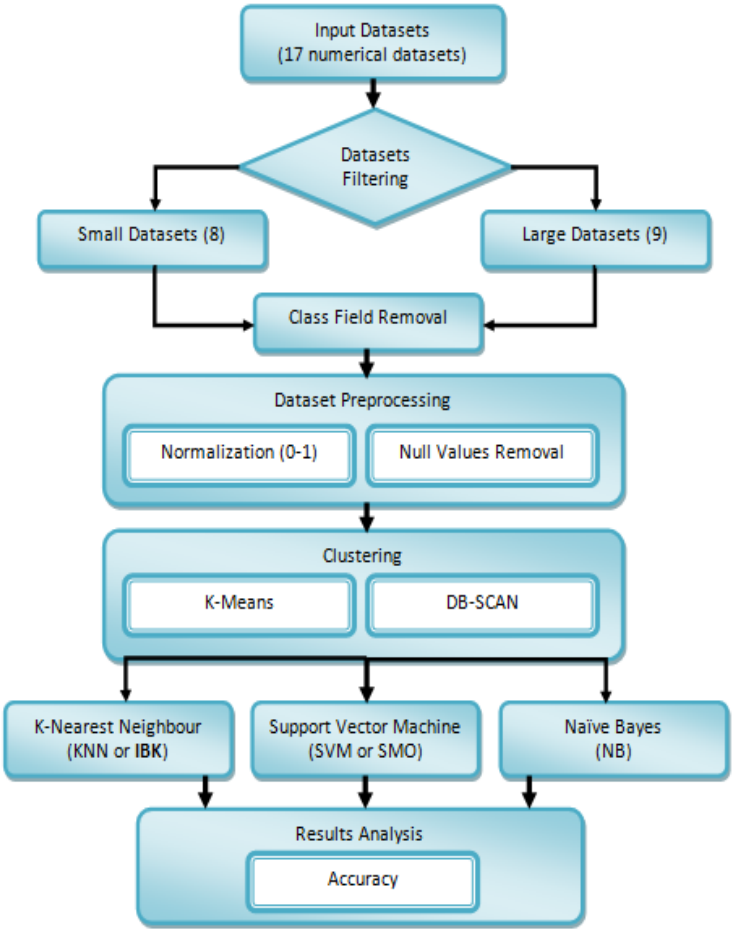

Figure 1: Proposed Methodology [24]

The detailed description of above-mentioned proposed methodology has been described below.

3.1. Input Datasets

For classification and regression, online machine learning dataset repositories offer many small and large datasets. Popular repositories include Kaggle, UCI machine learning repository, Keel, and LionBridge. Different academics use these internet databases for investigative studies and exploratory analysis. We used these repositories to retrieve 8 small and 9 large ML datasets for our investigation. Our research examined the efficiency of two popular clustering methods, K-Means and DB-Scan, utilizing these datasets.

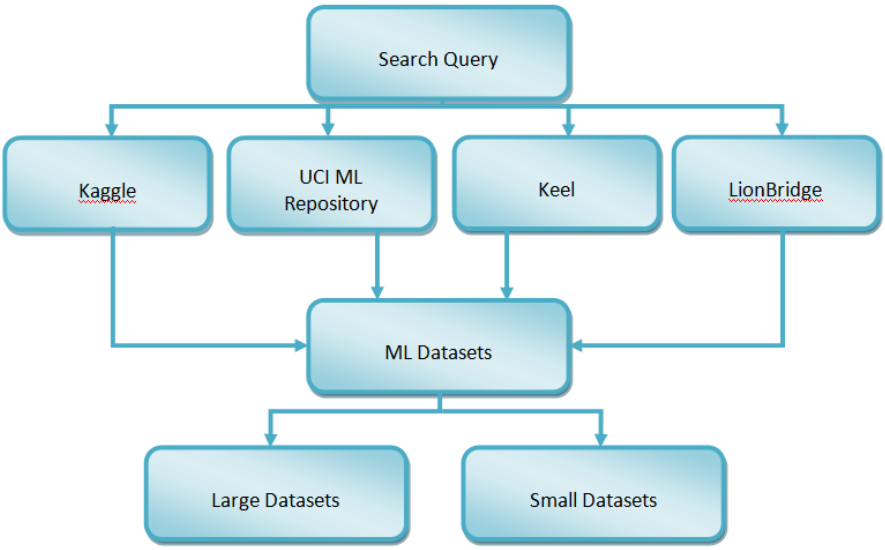

Figure 2: Datasets Searching and Filtering Process [25]

3.2. Dataset Filtering

After selecting 17 datasets, they are divided into small and large-scale datasets. This category is based on dataset instances. Small datasets have fewer than 1000 occurrences, while large datasets have more than 1000. The goal is to compare two clustering methods on large and small datasets. The figure above displays dataset collection and filtering.

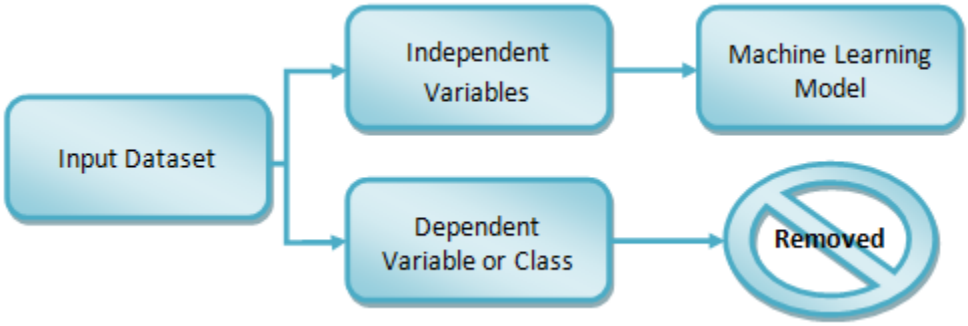

Figure 3: Class Field Removal [26]

3.3. Data Preprocessing

The use of public databases in research investigations is typically plagued by noise and missing numbers. A high-quality, noise-free dataset also boosts ML model efficiency. As we know, input datasets might be numerical, image-based, or sound-based, and each type of noise requires a distinct data mining technique to enhance. Since we are working with numerical datasets, we only investigated preprocessing approaches

that are routinely used to refine and improve numerical datasets. The table below lists two preprocessing methods we used in our investigation.

Table 1: Preprocessing Techniques

Preprocessing technique	Description
Normalization	Various data normalization techniques are used to normalize input datasets to (0-1) or (-1, 1). Min-Max, Z, and unit vector normalization are standard data or feature normalizing methods. Our study standardized input dataset numerical or quantitative attributes from 0 to 1 using min-max normalization [27].
Null Values Removal	As stated, missing values are a key concern in machine learning datasets. Common missing values removal methods include deleting rows or instances with missing data, replacing null values with column mean, median, or mode, assigning a specific value to all null cells, or using machine learning algorithms that support missing data. We used Weka's Remove-with-Filter to remove missing values from input datasets.[28]

3.4. Clustering

Data clustering is commonly used in machine learning to classify input information into two or more groups based on related properties. After data is divided into many groups or classes, each group is allocated a label. As previously stated, the goal of our research is to evaluate the efficiency of two well-known clustering techniques, DB-SCAN and K-Means. As a result, after preprocessing input datasets, these datasets are fed to the two clustering algorithms mentioned above. The number of clusters is set to two, dividing each dataset into two major categories depending on its features.

3.5. Classification

After applying clustering methods and then dataset splitting, the clustered training data is given into the classifier to determine how successfully the clustering algorithms classified the input datasets. In our study, we used three well-known machine learning classifiers: K-Nearest Neighbour (KNN), also known as Instance Based Classifier (IBK) in Weka, Support Vector Machine (SVM), and Naïve Bayes (NB). These classifiers are first trained on 70% of the training data and then tested on 30% of the testing data to determine their generalizability and the efficiency of the two clustering algorithms.

3.6. Results Analysis

There are several measures used to evaluate ML classifier performance. Accuracy, Precision, Recall, True Positives, False Positives, AUC, ROC, and others are popular evaluation measures. We used accuracy to evaluate classifier performance in our study.

The figure below illustrates a comprehensive graphical description of these steps.

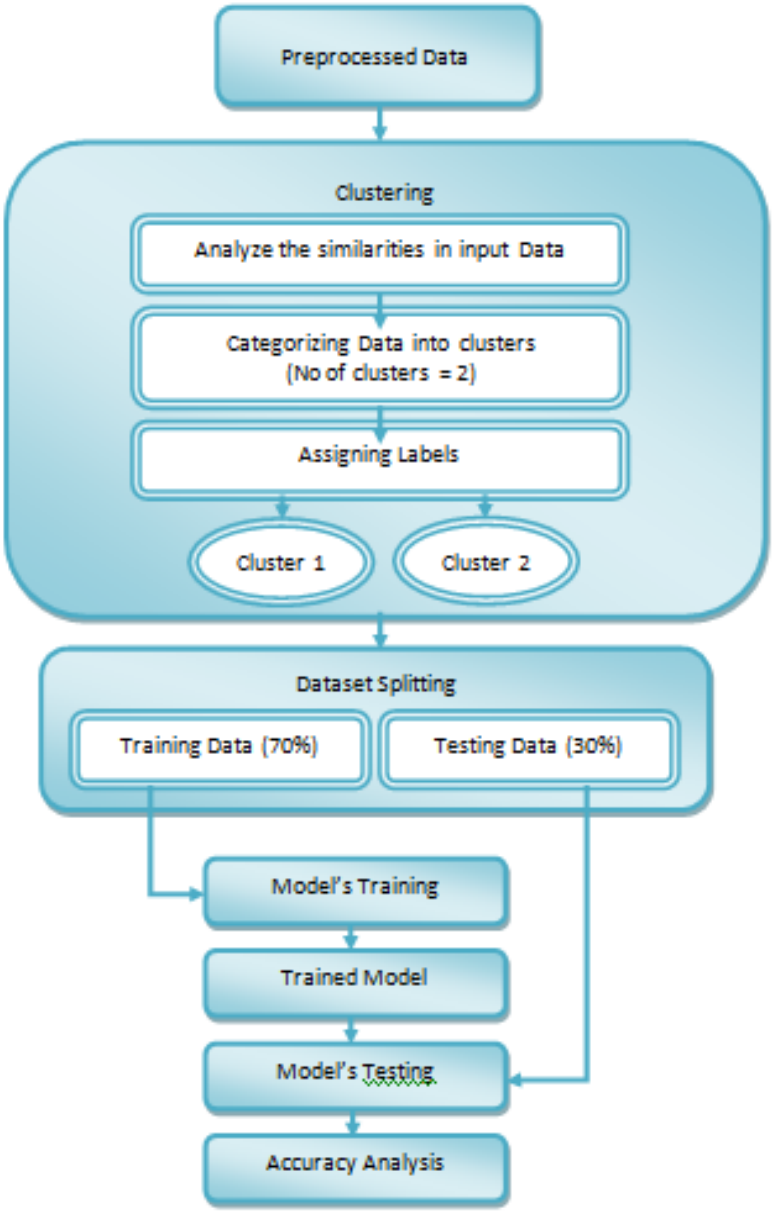

Figure 4: Detailed Model's Training Process [16]

4. Implementation and Results

In this Research, Weka 3.9.4 is used to evaluate dataset accuracy using clustering and classifiers. Data mining software Weka incorporates Machine Learning Algorithms. These algorithms can be applied on data or called from Java code. We used diverse datasets to compare K-Mean and DBSCAN clustering methods. These datasets are grouped using two algorithms, then three classifiers are deployed to evaluate clustering techniques. Results of experiments are below.

4.1. Results Analysis over Small Datasets

Below, for each dataset, are the accuracy graphs and the performance analysis (i.e., accuracy-wise) of the chosen clustering algorithms.

4.1.1. Autism Dataset

In the first scenario, we utilized the two clustering techniques previously mentioned to cluster the Autism Dataset. The table 2 below illustrates the dataset's description.

Table 2: Description of Autism Dataset

Dataset Characteristics	Value
No of Rows in Dataset	704
No of Columns in Dataset	21
Data Type of Attributes	Integer
Dataset Type	Classification
Containing Missing or Null Values	Yes

For performance analysis, the instance-based classifier (IBK), also referred to as KNN, Naïve Bayes, and SMO, is supplied the clustered dataset. Figure 16 illustrates the outcomes of the two clustering algorithms on this dataset.

Based on the results analysis, we have determined that the instance-based classifier, also known as KNN, exhibited the lowest accuracy among the two clustering algorithms. Conversely, the SMO or Support Vector Machine demonstrated the most effective performance among the two selected clustering techniques. Furthermore, the results of the experiments conducted indicate that the performance of DBSCAN is superior to that of the K-mean clustering algorithm on the Autism dataset, regardless of the classifier used.

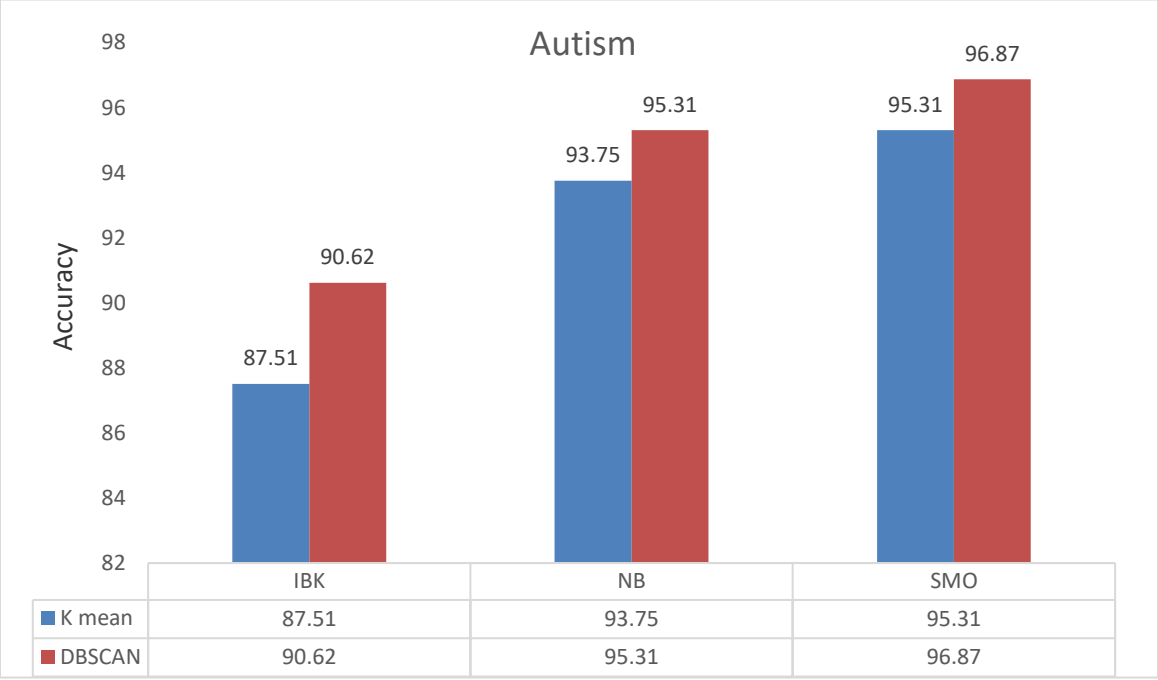

Figure 5: Performance on Autism Dataset

4.1.2. Breast Cancer Wisconsin Dataset

The second dataset that we have employed in our research is the breast cancer dataset, which is frequently employed for the automated diagnosis of breast cancer using machine learning-based techniques.

This dataset is primarily derived from digital breast images, which provide information about the characteristics of tumors or cell nuclei. As indicated in Table 3, this dataset is described below.

Table 3: Description of Breast Cancer Wisconsin Dataset

Dataset Characteristics	Value
No of Rows in Dataset	569
No of Columns in Dataset	32
Data Type of Attributes	Real
Dataset Type	Classification
Containing Missing or Null Values	No

In case of breast cancer dataset, a little variance has been noticed in accuracy of clustering techniques i.e., K-means algorithm has shown better results on two classifiers (including IBK and NB), while in case of SMO, DB-Scan has shown best accuracy of 96.15. In addition to this Naïve Bayes has depicted lowest accuracy of 73.01% on DB-Scan clustering algorithm.

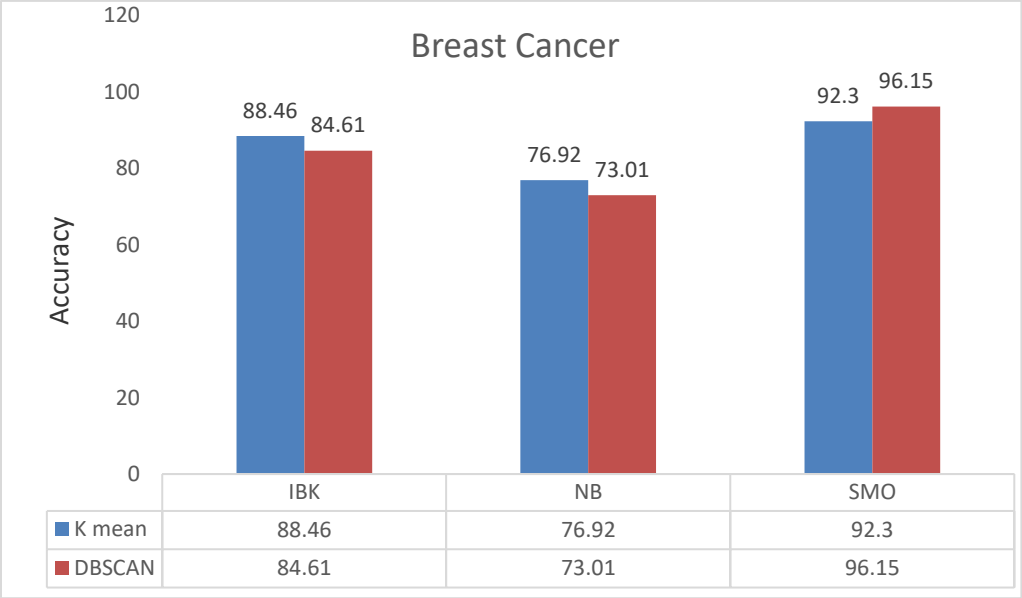

Figure 6: Performance on Breast Cancer Dataset

4.1.3. Contact Lenses Dataset

Our research has also employed an additional small-scale dataset to evaluate the effectiveness of clustering techniques on extremely small datasets. This dataset is accurate and comprehensive, as it contains no missing values. The primary objective of this dataset is to determine whether a patient requires contact lenses and whether they should be soft or firm. It comprises a total of three classes. The summary below contains additional details regarding this dataset.

Table 4: Description of Lenses Dataset

Dataset Characteristics	Value
No of Rows in Dataset	24
No of Columns in Dataset	4

Data Type of Attributes	Categorical
Dataset Type	Classification
Containing Missing or Null Values	No

The graph below illustrates the outcomes of the experiments that were conducted on this dataset.

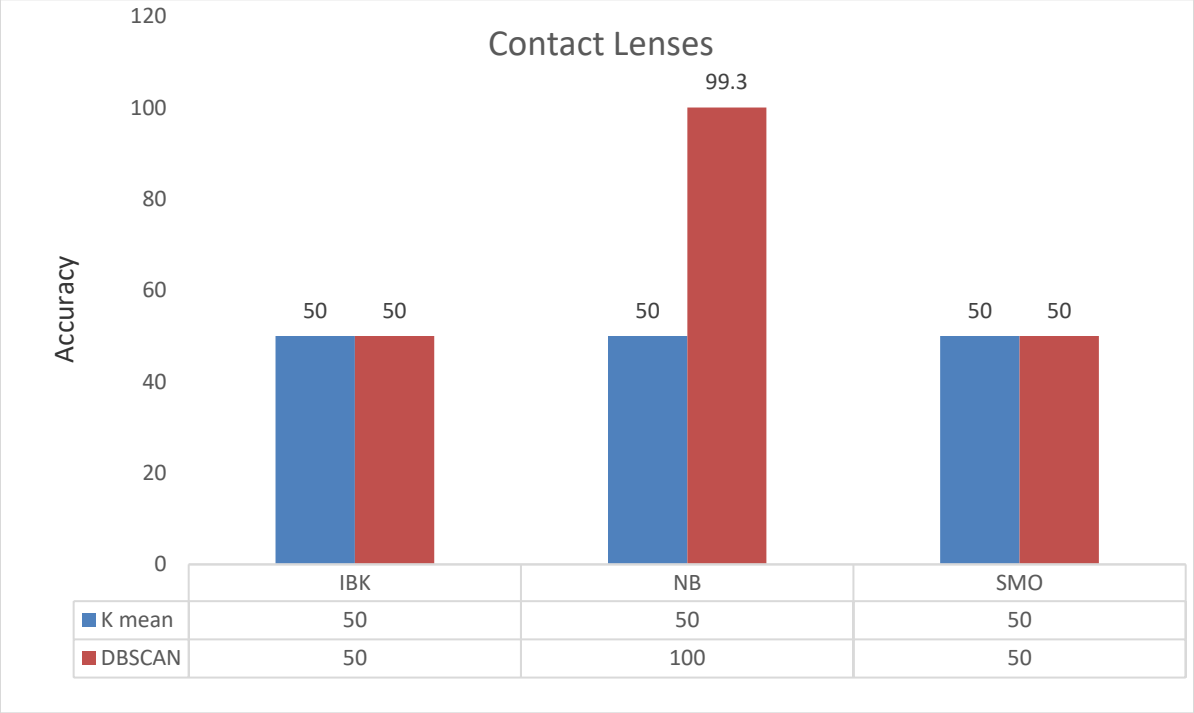

Figure 7: Performance on Contact Lenses Dataset

The classification models have not generalised well on this dataset, as evidenced by the experimental results. The primary explanation for this phenomenon may be the dataset's diminutive size. Naïve Bayes DBSCAN has demonstrated an accuracy of 99.3%, while IBK and SMO models have obtained 50% accuracy in both clustering techniques.

4.1.4. Diabetes Dataset

This dataset was compiled from two distinct sources: 1) paper documents and 2) electronic data recording devices. Various researchers have extensively employed this dataset to automate the diagnosis of diabetes disease using a variety of machine learning-based techniques. The table below provides a more detailed description of this dataset.

Table 5: Description of Diabetes Dataset

Dataset Characteristics	Value
No of Rows in Dataset	768
No of Columns in Dataset	20
Data Type of Attributes	Categorical and Integer
Dataset Type	Classification
Containing Missing or Null Values	No

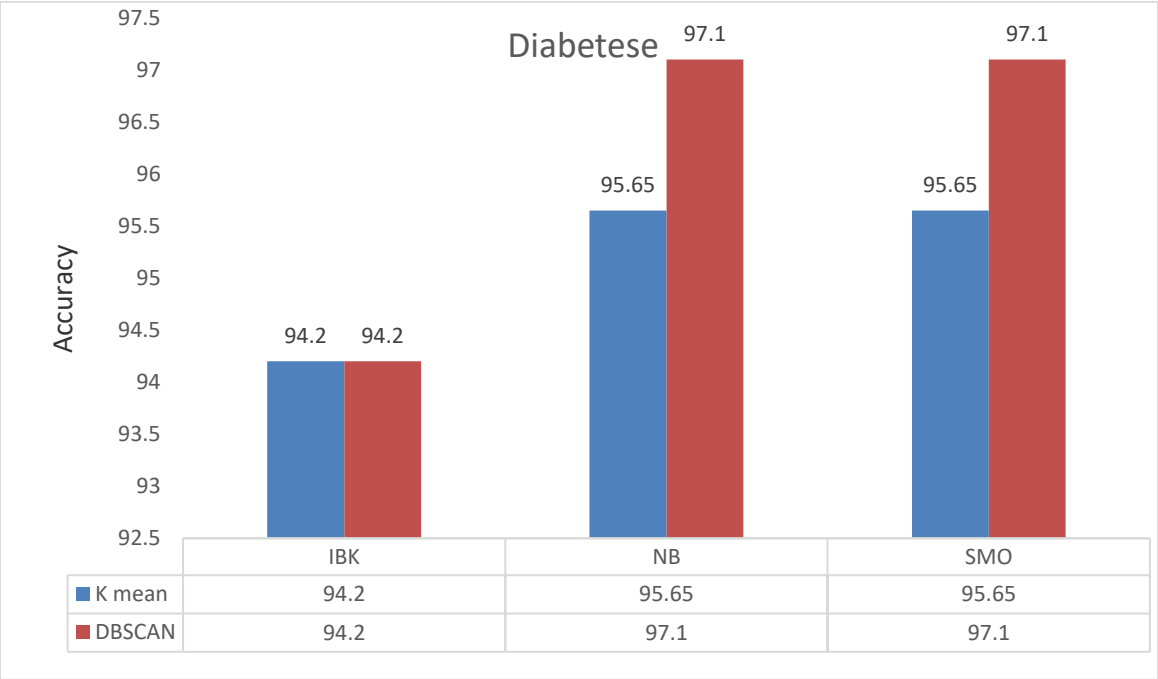

Figure 8: Performance on Diabetes Dataset

Upon analyzing the results, it is evident that both clustering techniques have demonstrated comparable results in the case of IBK (i.e., accuracy=94.2%). However, DBSCAN has outperformed K-Mean in the case of the other two classifiers, Naïve Bayes and SMO, achieving an accuracy of 97.1%

4.1.5. Titanic Dataset

Titanic dataset is an additional dataset that we have implemented in our investigation. The primary objective of this dataset is to forecast the likelihood of passenger survival aboard the Titanic. This dataset includes two classifications (YES and NO), which indicate whether the passenger has survived or not. Nine distinct risk factors have been employed to predict survival. The table below contains additional details regarding this dataset.

Table 6: Description of Titanic Dataset

Dataset Characteristics	**Value**
No of Rows in Dataset	891
No of Columns in Dataset	8
Data Type of Attributes	Categorical and Float
Dataset Type	Classification
Containing Missing or Null Values	Yes

Figure below illustrates the outcomes of the experiments conducted on the Titanic dataset. Regardless of the classifier employed, K-Mean has consistently obtained a maximum accuracy of 100%. This can be analyzed. Nevertheless, in the case of DBSCAN, IBK and NB have yielded identical results; however, SMO has attained a slightly lower level of accuracy, specifically 98.98%. In general, it is possible to infer that K-Means have demonstrated the most superior performance among all classifiers on the Titanic dataset.

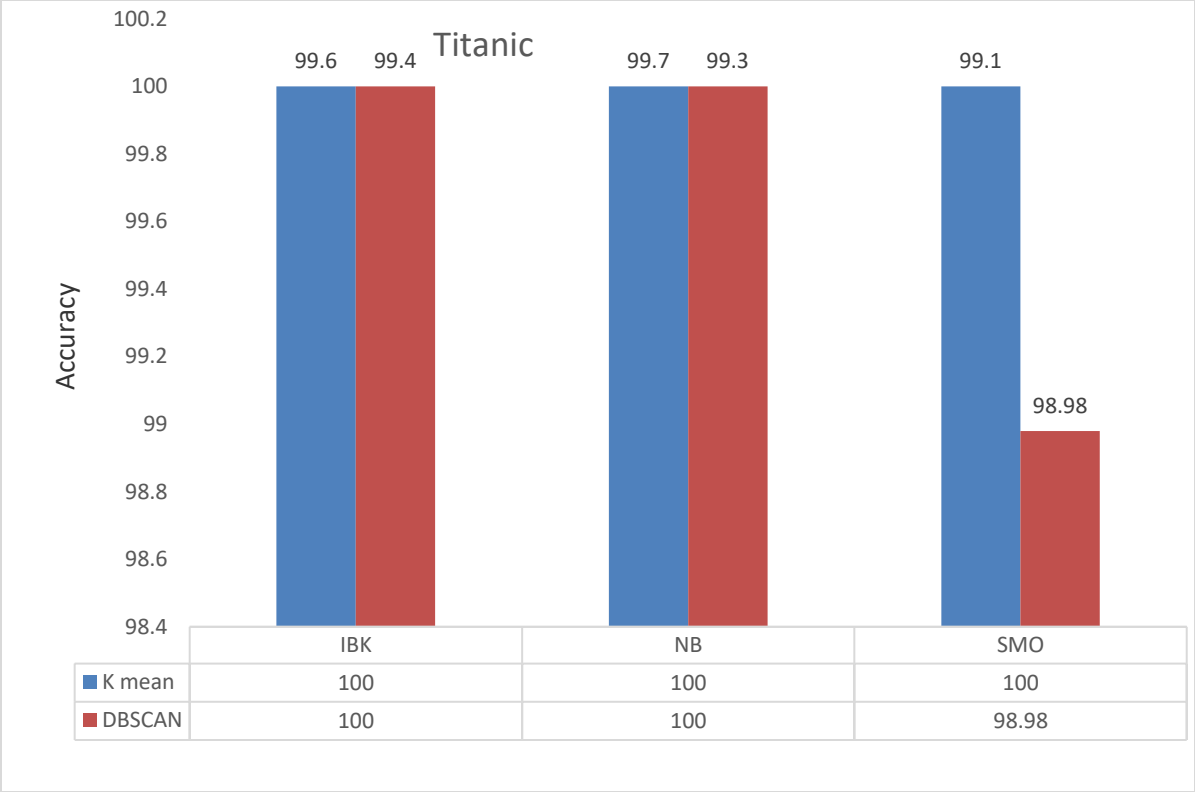

Figure 9: Performance on Titanic Dataset

4.1.6. Labor Dataset

The labor dataset is an additional dataset of limited extent that we have implemented in our investigation. This dataset has been previously employed in the literature to differentiate or categories unacceptable and acceptable contracts based on specific project attributes, such as wage, living allowance, and working hours. The table below provides a more detailed description of this dataset.

Table 7: Description of Labor Dataset

Dataset Characteristics	Value
No of Rows in Dataset	57
No of Columns in Dataset	16
Data Type of Attributes	Real, Integer and Categorical
Dataset Type	Classification
Containing Missing or Null Values	No

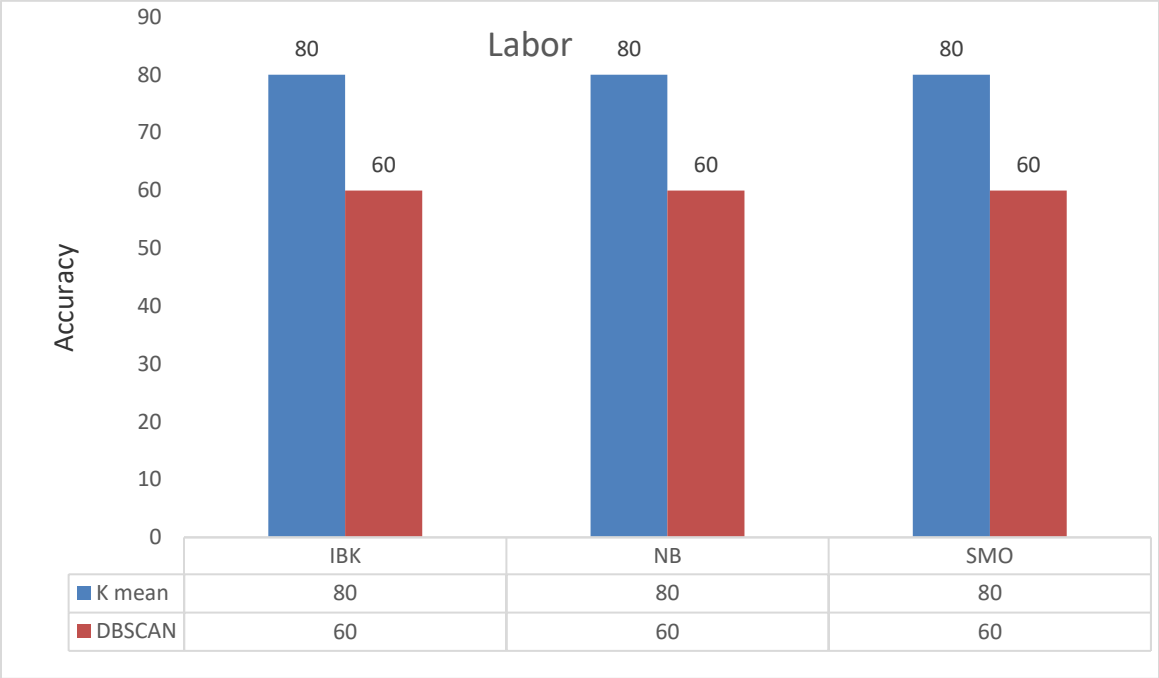

Figure 10: Performance on Labor Dataset

The labor dataset is insufficiently large to enhance the generalizability of ML classifiers. The results of the experiment conducted on this dataset indicate that the performance of K-Means is significantly superior to that of DB-Scan, despite the fact that both clustering techniques have yielded identical results across all classifiers. The maximum performance of 80% was obtained by K-Mean over all classifiers (i.e., IBK, NB, and SMO), while DB-Scan has achieved an accuracy of 60%.

4.1.7. Vote Dataset

Vote data from U.S. houses, which are emblematic of congressmen, is included in this dataset. These ballots are classified into nine distinct categories, including paired for, voted for, voted against, and paired against. Seventeen distinct categorical attributes comprise this dataset. The table below provides additional information regarding this dataset.

Table 8: Description of Vote Dataset

Dataset Characteristics	**Value**
No of Rows in Dataset	435
No of Columns in Dataset	16
Data Type of Attributes	Categorical
Dataset Type	Classification
Containing Missing or Null Values	Yes

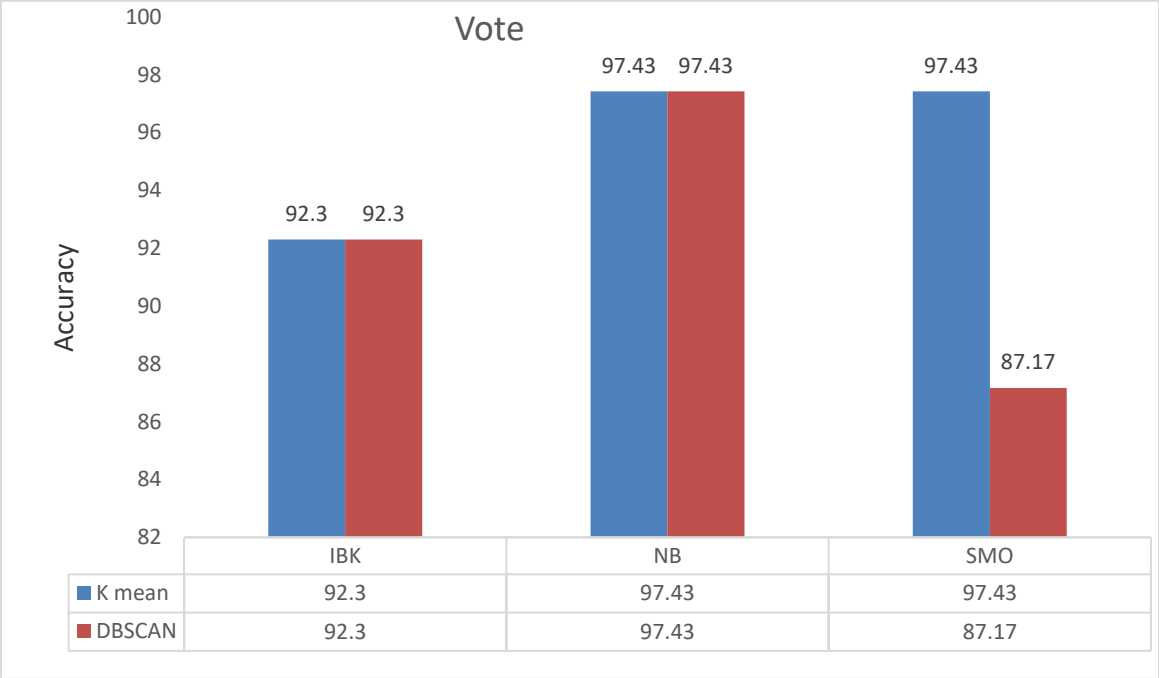

Figure 11: Performance on Vote Dataset

In the case of IBK and NB, both clustering algorithms have neared the same accuracy (92.3% and 97.43%). However, SMOK-Means have outperformed DB-Scan, with K-Means achieving an accuracy of 97.43% and DB-Scan achieving 87.17%, respectively. This is illustrated in the results graph. It is possible to infer that K-Means outperformed DB-Scan on the Vote dataset on a majority basis.

4.1.8. Soybean Dataset

The soybean dataset is the most recent dataset of a modest size that we have employed in our research. While only the first 15 classes have been utilized in prior research, this dataset contains a total of 19 classes. The class imbalance issue is caused by the relatively low number of instances pertaining to the subsequent four classes, which is why they are not being utilized. The table below provides a more detailed description of this dataset.

Table 9: Description of Soybean Dataset

Dataset Characteristics	**Value**
No of Rows in Dataset	307
No of Columns in Dataset	35
Data Type of Attributes	Categorical
Dataset Type	Classification
Containing Missing or Null Values	Yes

The graph below plainly demonstrates that DB-Scan has outperformed the two clustering techniques that were specified in all of the experiments. The highest accuracy of 98.3% was obtained by DB-Scan over the NB classifier, while the lowest accuracy was achieved over the IBK classifier at 91.8%. In summary, it could be concluded that the Soybean dataset has yielded superior results when compared to DB-Scan.

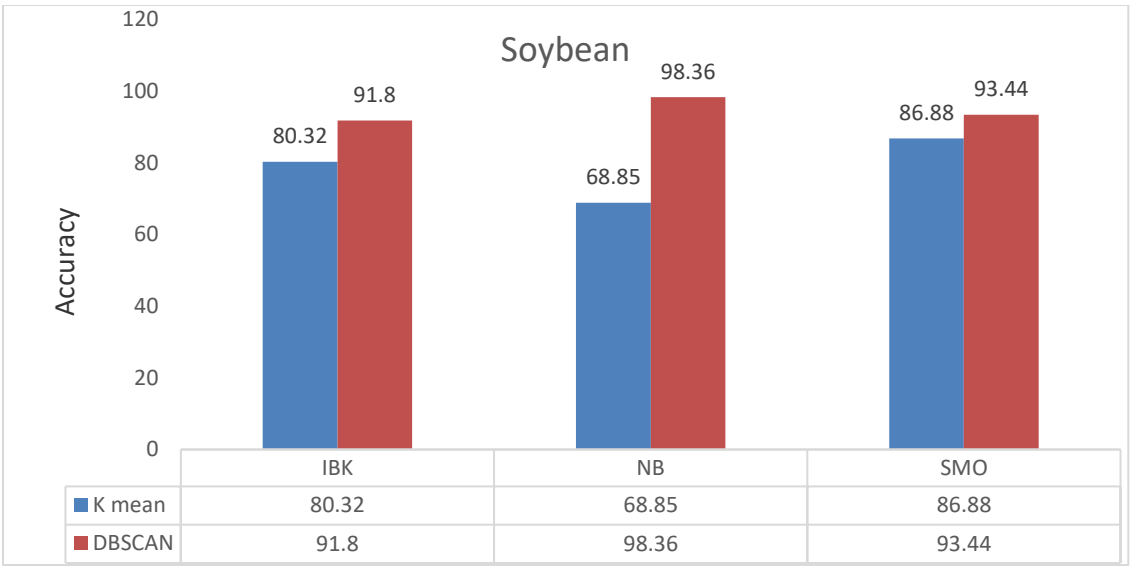

Figure 12: Performance over Soybean Dataset

4.1.9. Analysis of Small Datasets

We have conducted a collective analysis, as illustrated in the table below, to evaluate the collective impact or performance of clustering algorithms over small datasets. Specifically, we are interested in determining which clustering technique performs better when combined with which classifier.

Table 10: Analysis of Small Datasets

Small Datasets	K-Means	DB-Scan	Classifier
Autism		✓	SMO
Breast Cancer		✓	SMO
Lenses		✓	NB
Diabetes		✓	SMO, NB
Titanic	✓		SMO, NB. IBK
Labor	✓		SMO, NB. IBK
Vote	✓		SMO, NB
Soybean		✓	NB

Based on the examination of the aforementioned results, it is evident that DB-Scan has outperformed the K-Means clustering technique in the majority of instances. Specifically, K-Means has outperformed DB-Scan on three datasets, while DB-Scan has outperformed K-Means on five datasets. In addition, we have conducted an analysis to determine which classifier has yielded the most favorable outcomes when implemented with the most effective clustering technique. According to this analysis, NB and SMO have achieved the highest accuracy across six datasets, while IBK has only achieved the highest accuracy across two datasets.

4.2. Results Analysis over Large Datasets

4.2.1. Ring Dataset

The ring-norm dataset is the initial large-scale dataset that we have employed in our research. It is a 20-dimensional classification dataset that contains two classes. The summary below contains an additional description of this dataset.

Table 11: Description of Ring-norm Dataset

Dataset Characteristics	Value
No of Rows in Dataset	7400
No of Columns in Dataset	20
Data Type of Attributes	Integer, Nominal and Real
Dataset Type	Classification
Containing Missing or Null Values	No

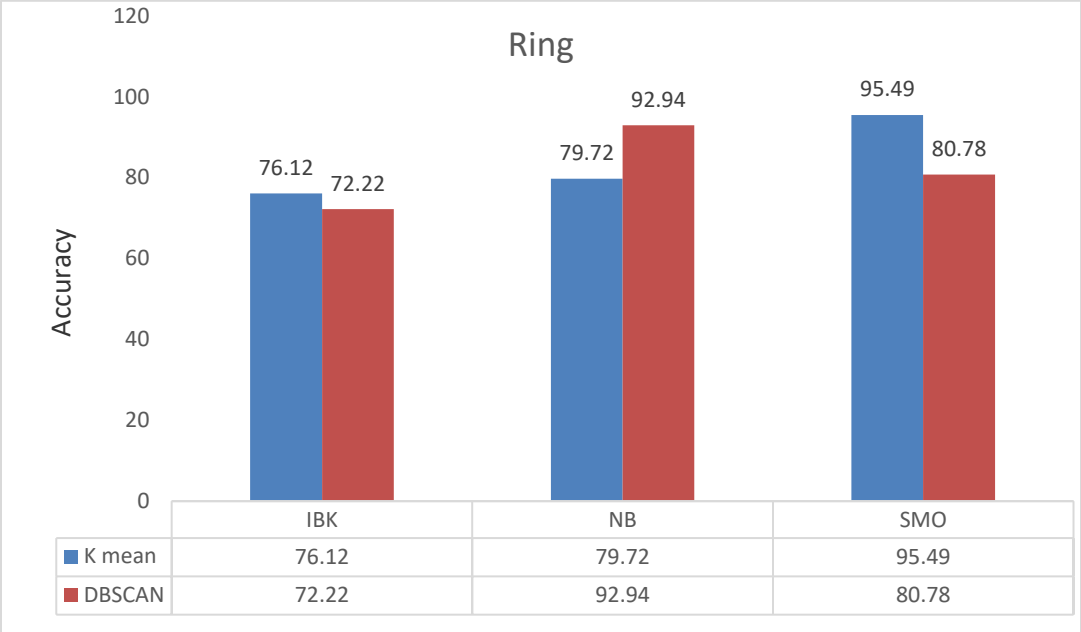

Figure 13: Performance on Ringnorm Dataset

The K-mean outperformed the ring-norm dataset in the majority of cases, as evidenced by the results analysis. Furthermore, the K-means clustering algorithm has demonstrated superior performance in the context of IBK and SMO classifiers, achieving a maximal accuracy of 95.49%. Conversely, the NB DB-Scan algorithm has outperformed the former, achieving an accuracy of 92.94%.

4.2.2. Magic Dataset

The Magic Gamma Telescope dataset is the second large-scale dataset that we have implemented in our investigation. It includes two classes: one for distinguishing gamma particles or signals from hadrons or background. Monte Carlo has made the dataset publicly available. The table below contains additional information regarding this dataset.

Table 12: Description of Magic Dataset

Dataset Characteristics	Value
No of Rows in Dataset	19020
No of Columns in Dataset	11
Data Type of Attributes	Real
Dataset Type	Classification
Containing Missing or Null Values	No

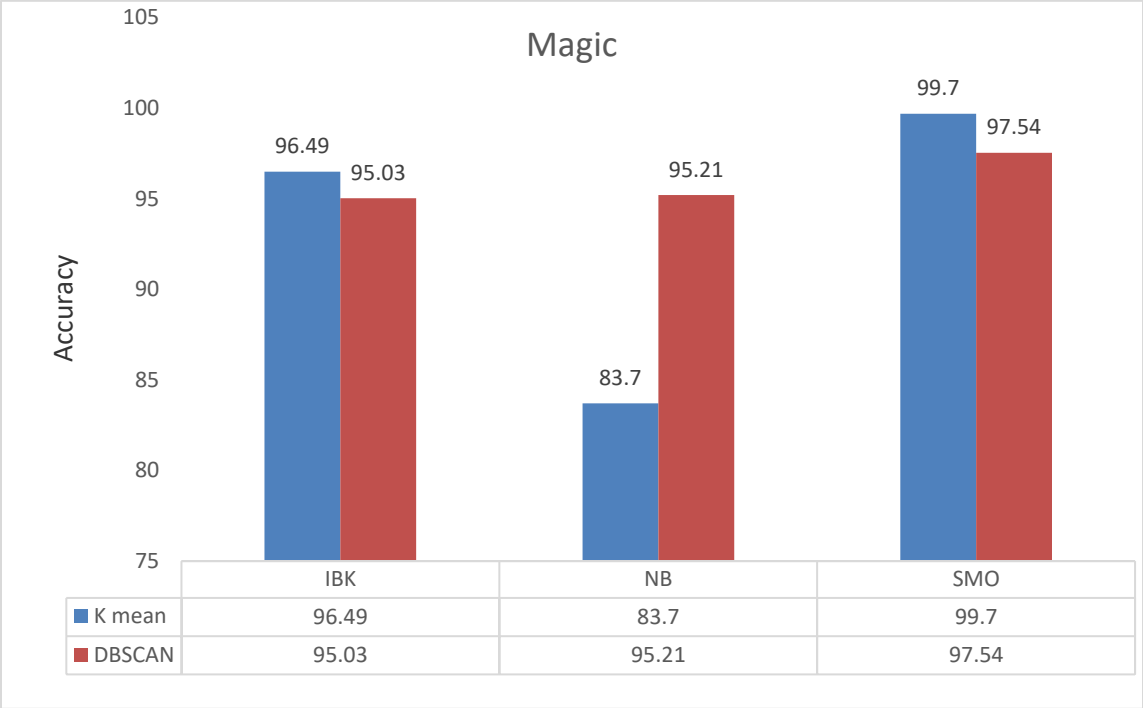

Figure 14: Performance over Magic Dataset

The experimental results reveal that K-means outperformed the two clustering strategies under consideration. K-means outperformed two of the three classifiers, IBK and SMO, with a maximum accuracy of 99.7%, although NB DB-Scan outperformed the K-mean algorithm.

4.2.3. Wine Quality Dataset

The wine quality dataset is the third large-scale dataset that we have employed in our experimental phase. The primary objective of this dataset is to evaluate the quality of wine. To achieve this, it has been compiled from two distinct varieties of Portuguese wine: red and white. This dataset has the potential to be employed to address both classification and regression issues. The quality grade is assigned within the range of 0 to 10. The summary below contains a more detailed description of this dataset.

Table 13: Description of Wine Quality Dataset

Dataset Characteristics	**Value**
No of Rows in Dataset	4898
No of Columns in Dataset	12
Data Type of Attributes	Real
Dataset Type	Regression, Classification
Containing Missing or Null Values	N/A

Again, K-Means has outperformed DB-SCAN in combination with two classifiers, NB and SMO, in the case of Wine quality in a publicly available large-scale dataset. SMO has attained the highest accuracy of 98.41% in the K-Means clustering technique. Nevertheless, DB-SCAN has outperformed K-Means in the context of the IBK classifier, achieving an accuracy of 97.73%.

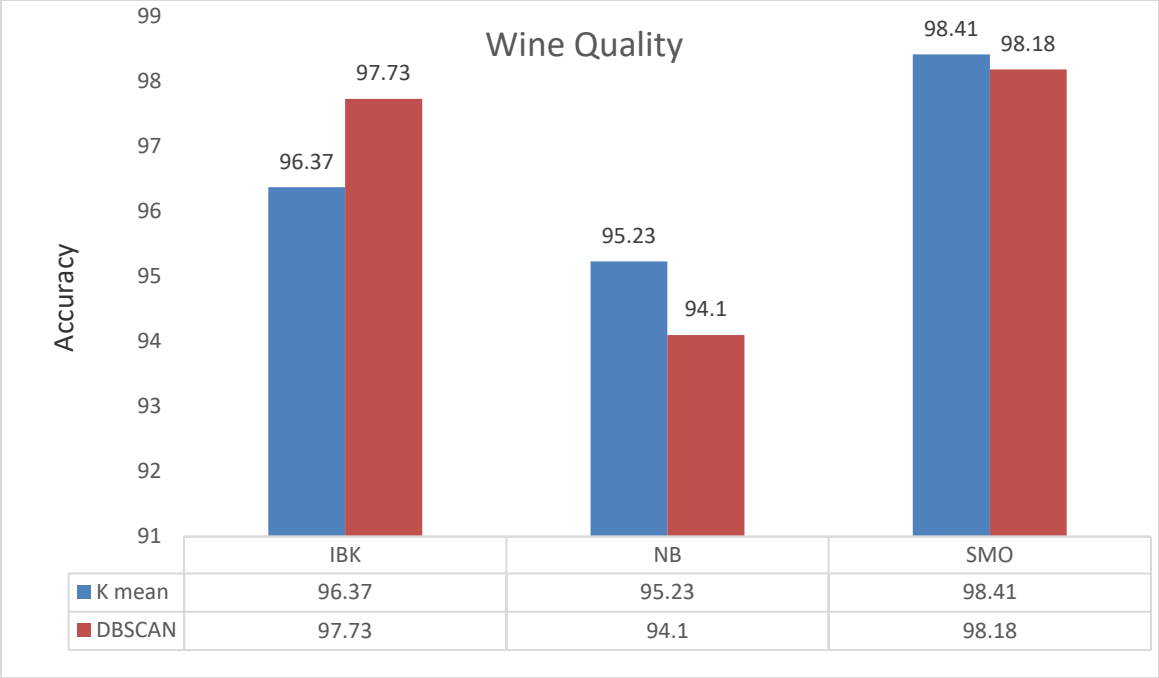

Figure 15: Performance over Win Quality Dataset

4.2.4. Shuttle Dataset

Shuttle, which is also known as the Statlog dataset, is another publicly accessible large-scale dataset. This dataset is primarily used for classification purposes and comprises a total of seven classes, which were arranged in chronological order in the original dataset. One of the seven classes is highly imbalanced, resulting in an accuracy of 80%. Consequently, the primary objective is to obtain a performance within the range of 90 to 90.9%. The summary below contains additional information regarding this dataset.

Table 14: Description of Shuttle Dataset

Dataset Characteristics	Value
No of Rows in Dataset	4898
No of Columns in Dataset	12
Data Type of Attributes	Real
Dataset Type	Regression, Classification
Containing Missing or Null Values	N/A

Similar to previous experiments, the shuttle dataset has also demonstrated superior results from two of the three classifiers, namely IBK and SMO, in comparison to the K-Mean clustering algorithm, with a maximal accuracy of 99.21%. In the case of NB, DB-SCAN has outperformed K-Means and has attained an accuracy of 93.77%.

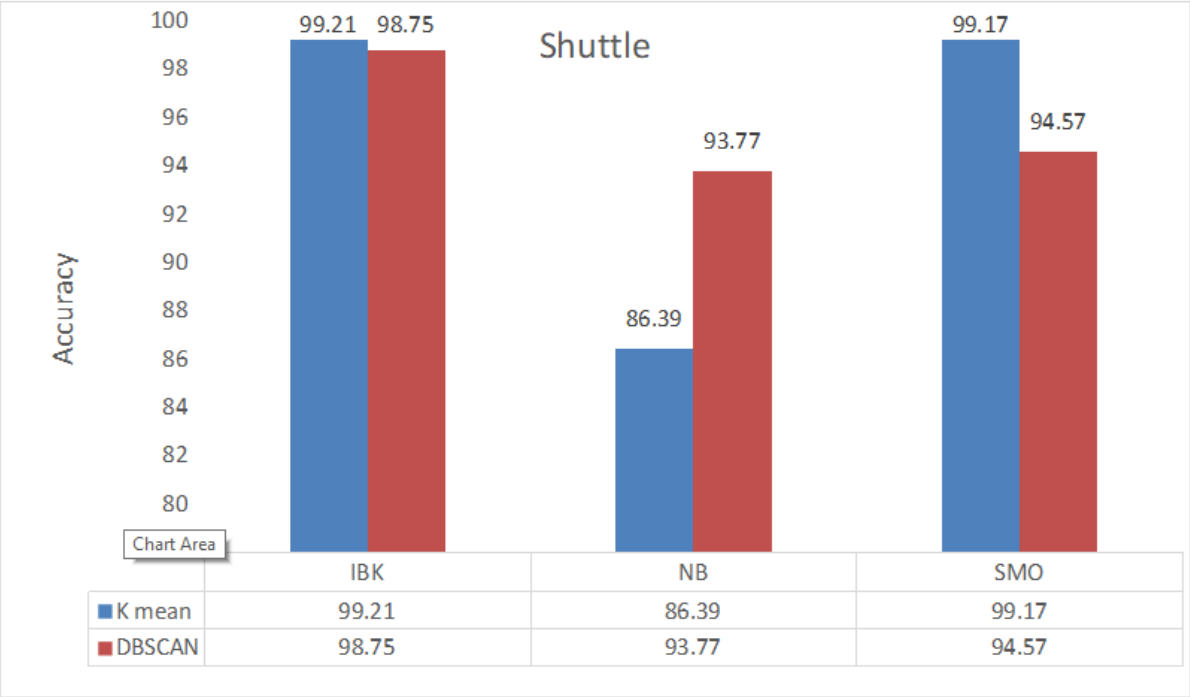

Figure 16: Performance over Shuttle Dataset

4.2.5. Thyroid Dataset

The main aim of this dataset is to differentiate between individuals with thyroid disease and those who are not. Additionally, two additional variations of this dataset have been made available to the public. The table below contains the primary attributes of this dataset.

Table 15: Description of Thyroid Dataset

Dataset Characteristics	**Value**
No of Rows in Dataset	7200
No of Columns in Dataset	21
Data Type of Attributes	Real and Categorical
Dataset Type	Classification
Containing Missing or Null Values	N/A

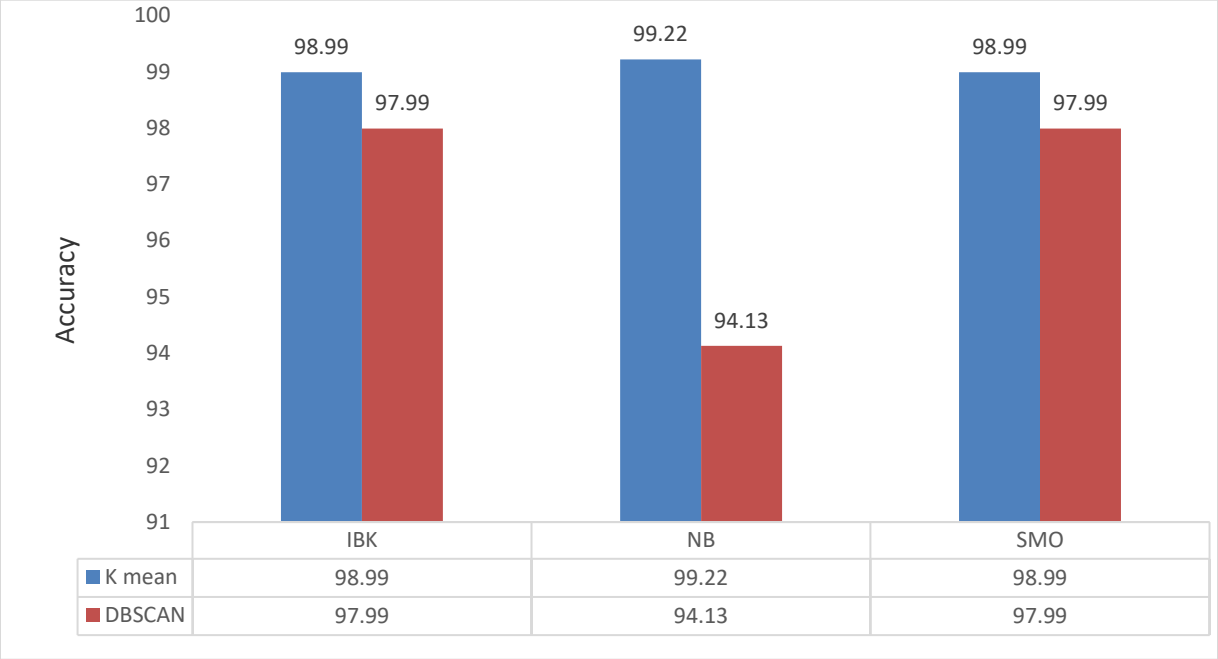

Figure 17: Performance on Thyroid Dataset

For the Thyroid dataset, all three classifiers (IBK, NB, and SMO) have demonstrated superior performance in comparison to the K-Mean clustering technique. Additionally, IBK and SMO have attained an accuracy of 98.99% in the K-Means clustering technique.

4.2.6. Texture Dataset

The texture dataset is a 40-dimensional dataset that spans a large scale and includes 11 distinct classes. Table below provides additional information regarding this dataset.

Table 16: Description of Texture Dataset

Dataset Characteristics	**Value**
No of Rows in Dataset	5500
No of Columns in Dataset	40
Data Type of Attributes	Real
Dataset Type	Classification
Containing Missing or Null Values	No

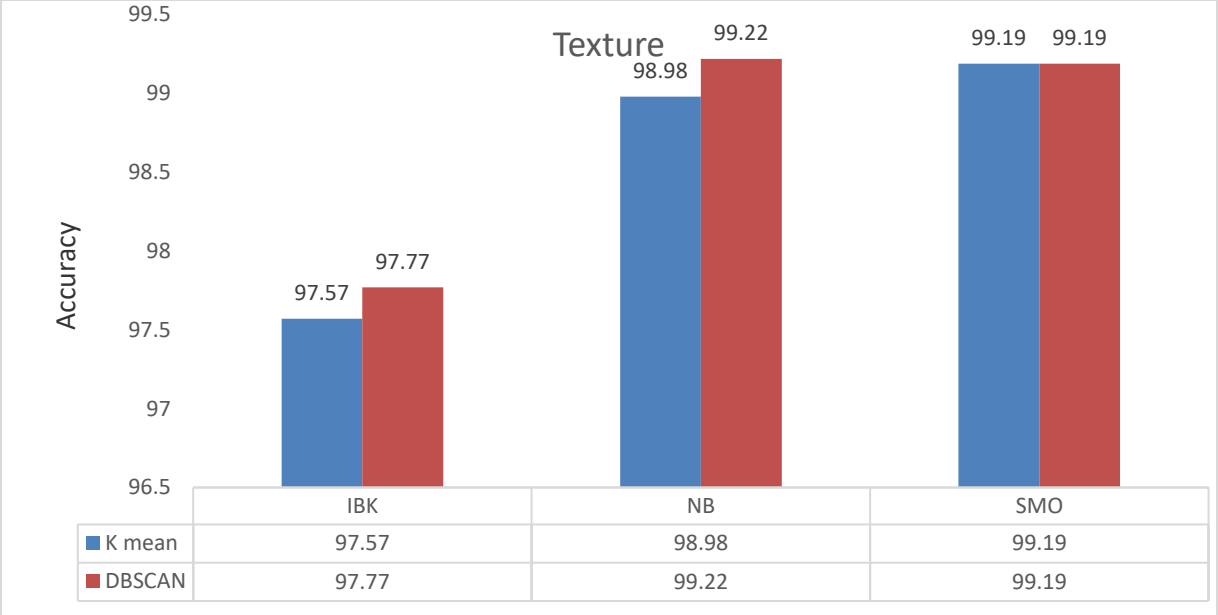

Figure 18: Performance over Texture Dataset

In contrast to previous experiments, the texture dataset has yielded the most favorable results with DB-SCAN. Additionally, both clustering techniques have attained an accuracy of 99.19% when employing SMO. DB-SCAN has obtained the highest accuracy in the case of NB and IBK, whereas NB has approached the maximum accuracy of 100%.

4.2.7. Marketing Dataset

This dataset was collected from marketing campaigns that were conducted by Portuguese financial institutes. Phone communications have been implemented during these campaigns. The primary objective of this dataset is to determine whether a client intends to enroll in a term deposit. Table below delineates additional attributes of this dataset.

Table 17: Description of Marketing Datset

Dataset Characteristics	**Value**
No of Rows in Dataset	45211
No of Columns in Dataset	17
Data Type of Attributes	Real
Dataset Type	Classification
Containing Missing or Null Values	N/A

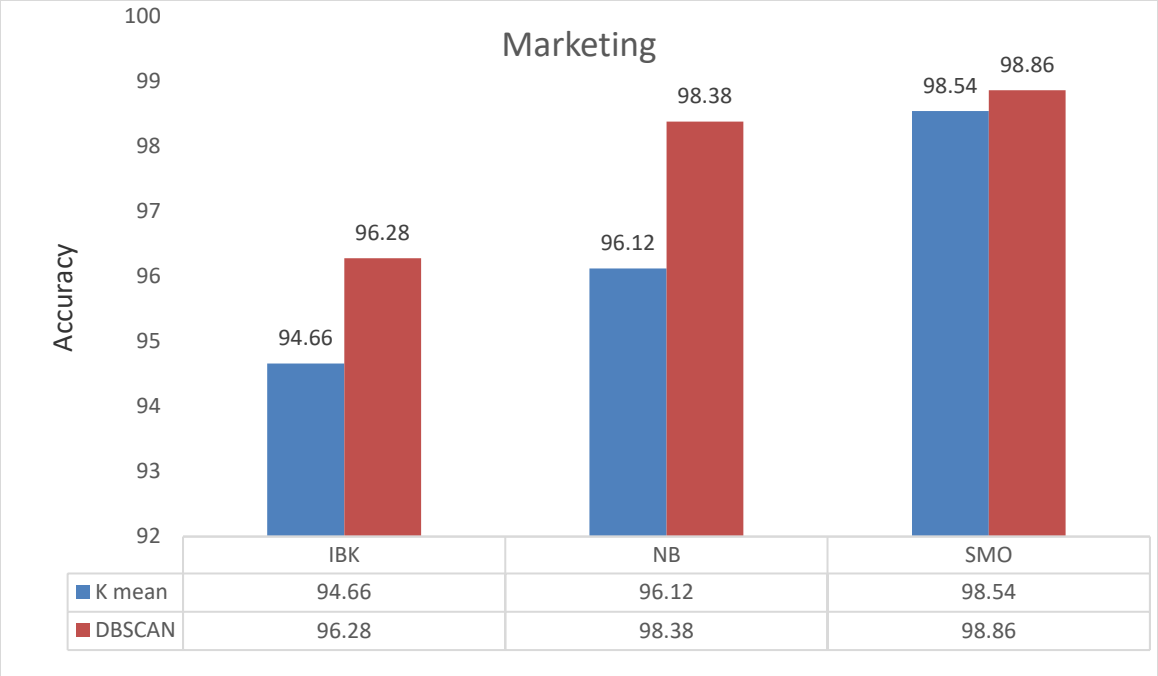

Figure 19: Performance over Marketing Dataset

The DB-SCAN clustering algorithm has demonstrated the most effective performance in the Marketing Dataset when combined with all three of the selected classifiers. SMO has obtained the highest accuracy (98.86%) when compared to data clustered by DBSCAN. IBK has attained the lowest accuracy (94.66%) when combined with K-Mean.

4.2.8. Letter Dataset

The letter recognition dataset is the second-to-last large-scale dataset that we have implemented in our investigation. The primary objective or objective of this dataset is to identify English alphabets from a rectangular grid of black and white pixels. The 20,000 distinct instances of this dataset are generated by randomly distorting the alphabetical images of 20 distinct font styles. The summary below contains a few of the dataset's most significant attributes.

Table 18: Description of Letters Dataset

Dataset Characteristics	**Value**
No of Rows in Dataset	20,000
No of Columns in Dataset	16
Data Type of Attributes	Integer
Dataset Type	Classification
Containing Missing or Null Values	No

Figure 20: Performance on Letter Dataset

DB-SCAN has obtained the highest accuracy on the letter's recognition dataset, as indicated by the experimental analysis. However, K-Means has outperformed DB-SCAN in the majority of cases. For example, K-Means outperforms DB-SCAN in the context of IBK and SMO, while DB-SCAN has obtained the highest accuracy of 98.88% in the context of NB.

4.2.9. Kr Vs K Dataset

The King-Rook Vs King-Pawn Kr-Vs-K dataset, also referred to as the chess dataset, is the most recent publicly available dataset that we have employed to evaluate the performance of clustering algorithms on large-scale datasets. The primary objective of this two-class dataset is to determine whether white individuals are capable of achieving victory. The table below provides a more detailed description of this dataset.

Table 19: Description of Kr Vs K Dataset

Dataset Characteristics	**Value**
No of Rows in Dataset	3196
No of Columns in Dataset	36
Data Type of Attributes	Categorical
Dataset Type	Classification
Containing Missing or Null Values	No

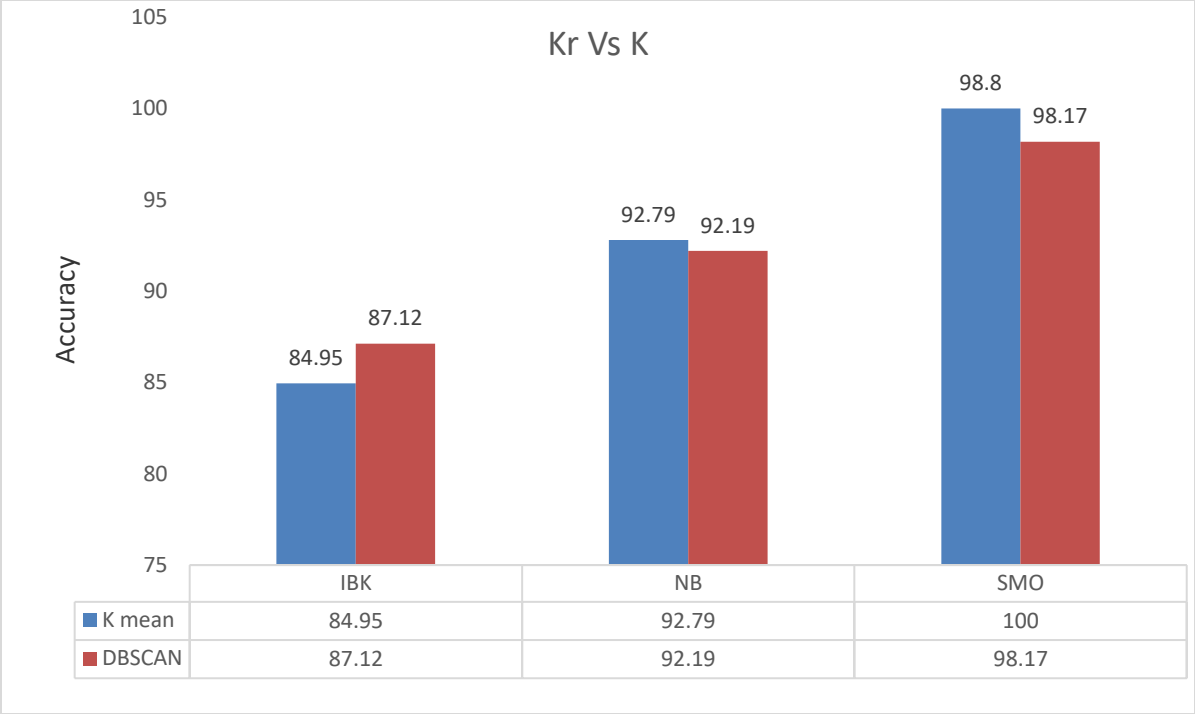

Figure 21: Performance on Kr Vs K Dataset

The efficacy of the K-Means clustering algorithm is superior to that of DB-SCAN in the case of NB and SMO classifiers, as illustrated in the graphic above. Nevertheless, the performance of DB-SCAN is superior to that of the K-Means algorithm when IBK is implemented for the classification of clustered data. As a result, it is possible to infer that the K-Means clustering technique is more accurate than the DB-SCAN clustering technique in the Kr-Vs-K dataset. Upon conducting an analysis of the classifiers, it was observed that SMO exhibited the highest level of performance among the three classifiers mentioned, with an accuracy of 100%.

4.2.10. Analysis of Large Datasets

Table 20: Results Analysis of Large Datasets

Large Datasets	K-Means	DB-Scan	Classifier
Ring-Norm Dataset	✓		SMO
Magic Dataset	✓		SMO
Wine Quality Dataset	✓		SMO
Shuttle Dataset	✓		IBK
Thyroid	✓		SMO, IBK
Texture		✓	NB
Marketing		✓	SMO
Letter		✓	NB
Kr-Vs-K	✓		SMO

Based on the comprehensive results analysis of experiments conducted over large datasets (i.e., as illustrated in Table 19), it is possible to infer that the K-Means algorithm outperforms the DB-SCAN algorithm in the context of large datasets. For example, of the nine large scale datasets utilized in our study,

K-Means outperformed DB-SCAN in the case of six datasets. Furthermore, SMO has demonstrated the highest level of accuracy in the context of datasets. Consequently, the clustering algorithm K-Means, when used in conjunction with SMO or SVM, yields superior outcomes for large-scale datasets.

Our study's limitations include the use of only 17 datasets, which may not represent the diversity of real-world data, and the focus on just two clustering algorithms, excluding many others that could yield different results. Additionally, we relied solely on accuracy for performance analysis, neglecting other important metrices.

5. Conclusion

Clustering, which groups related objects or datasets, is a popular unsupervised machine learning approach. These are clusters. various clusters of items have various features, and different similarity measures are used to compare them. Model-based, partitioning-based, hierarchal-based, grid-based, density-based, and constraint-based clustering methods are used in data analysis, image processing, pattern recognition, and market research. Given these ubiquitous clustering applications, finding the best efficient and accurate algorithm is vital. Publicly available small and big datasets are used to solve machine learning challenges and analyze algorithm performance. A large dataset is needed to fully train any machine learning algorithm, as models learned on larger datasets are more generalizable. However, most publicly accessible datasets are small or contain missing values. Small datasets often cause overfitting. Designing or developing adaptable algorithms that perform well on tiny datasets is crucial. We conducted an exploratory study to determine the best unsupervised machine learning clustering algorithm for small and large datasets. We chose two well-known clustering algorithms to test their performance in the case. The chosen clustering techniques are DB-SCAN and K-Means. To complete the study, 17 large and minor datasets were collected. Eight small and nine big datasets were preprocessed (normalized and null values eliminated). The two clustering methods receive data from the preprocessed dataset without class field. The clustered data is supplied to IBK, SVM, and NB machine learning classifiers for performance analysis. The final performance study of these algorithms used accuracy. According to the results, K-Means algorithms perform better on large datasets, whereas DB-SCAN performs better on small datasets.

In the future, we plan to broaden the scope of our analysis by incorporating additional clustering techniques into the research that has been carried out.

References

[1] Marino, Marina, and Cristina Tortora. "A comparison between K-means and Support Vector Clustering for Categorical Data." *Statistica applicata* 21, no. 1 (2009): 5-16.

[2] Namratha, M., and T. R. Prajwala. "A comprehensive overview of clustering algorithms in pattern recognition." *IOSR Journal of Computer Engineering* 4, no. 6 (2012): 23-30.

[3] Abualigah, Laith Mohammad, and Ahamad Tajudin Khader. "Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering." *The Journal of Supercomputing* 73 (2017): 4773-4795.

[4] Hinneburg, Alexander, and Daniel A. Keim. "Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering." (1999).

[5] Dubes, Richard, and Anil K. Jain. "Clustering techniques: the user's dilemma." *Pattern Recognition* 8, no. 4 (1976): 247-260.

[6] Huang, Shujun, Nianguang Cai, Pedro Penzuti Pacheco, Shavira Narrandes, Yang Wang, and Wayne Xu. "Applications of support vector machine (SVM) learning in cancer genomics." *Cancer genomics & proteomics* 15, no. 1 (2018): 41-51.

[7] Das, Amit Kumar, Saptarsi Goswami, Amlan Chakrabarti, and Basabi Chakraborty. "A new hybrid feature selection approach using feature association map for supervised and unsupervised classification." *Expert Systems with Applications* 88 (2017): 81-94.

[8] Abualigah, Laith Mohammad, Ahamad Tajudin Khader, and Essam Said Hanandeh. "A new feature selection method to improve the document clustering using particle swarm optimization algorithm." *Journal of Computational Science* 25 (2018): 456-466.

[9] Ettouil, Monia, Habib Smei, and Abderrazak Jemai. "Particle swarm optimization on fpga." In *2018 30th International Conference on Microelectronics (ICM)*, pp. 32-35. IEEE, 2018.

[10] Arevalillo-Herráez, Miguel, Aladdin Ayesh, Olga C. Santos, and Pablo Arnau-González. "Combining supervised and unsupervised learning to discover emotional classes." In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pp. 355-356. 2017.

[11] Ibrahim, Rehab Ali, Ahmed A. Ewees, Diego Oliva, Mohamed Abd Elaziz, and Songfeng Lu. "Improved salp swarm algorithm based on particle swarm optimization for feature selection." *Journal of Ambient Intelligence and Humanized Computing* 10 (2019): 3155-3169.

[12] Idris, Adnan, Muhammad Rizwan, and Asifullah Khan. "Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies." *Computers & Electrical Engineering* 38, no. 6 (2012): 1808-1819.

[13] Shokouhifar, Mohammad, and Ali Jalali. "Optimized sugeno fuzzy clustering algorithm for wireless sensor networks." *Engineering applications of artificial intelligence* 60 (2017): 16-25.

[14] Xue, Bing, Mengjie Zhang, Will N. Browne, and Xin Yao. "A survey on evolutionary computation approaches to feature selection." *IEEE Transactions on evolutionary computation* 20, no. 4 (2015): 606-626.

[15] Zhang, Degan, Hui Ge, Ting Zhang, Yu-Ya Cui, Xiaohuan Liu, and Guoqiang Mao. "New multi-hop clustering algorithm for vehicular ad hoc networks." *IEEE Transactions on Intelligent Transportation Systems* 20, no. 4 (2018): 1517-1530.

[16] Xiang, Wenkun, Hao Zhang, Rui Cui, Xing Chu, Keqin Li, and Wei Zhou. "Pavo: A RNN-based learned inverted index, supervised or unsupervised?." *IEEE Access* 7 (2018): 293-303.

[17] Hofmann, Thomas. "Unsupervised learning by probabilistic latent semantic analysis." *Machine learning* 42 (2001): 177-196.

[18] Mishra, Priya, Brijesh Raj Swain, and Aleena Swetapadma. "A review of cancer detection and prediction based on supervised and unsupervised learning techniques." *Smart healthcare analytics: state of the art* (2022): 21-30.

[19] Camastra, Francesco, Marco Spinetti, and Alessandro Vinciarelli. "Offline Cursive Character Challenge: a New Benchmark for Machine Learning and Pattern Recognition Algorithms." In *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 2, pp. 913-916. IEEE, 2006.

[20] Tilson, L. V., P. S. Excell, and R. J. Green. "A generalisation of the fuzzy c-means clustering algorithm." In *International Geoscience and Remote Sensing Symposium,'Remote Sensing: Moving Toward the 21st Century'.*, vol. 3, pp. 1783-1784. IEEE, 1988.

[21] Zhang, Huizhen, Fan Liu, Yuyang Zhou, and Ziying Zhang. "A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem." *Information Sciences* 539 (2020): 347-374.

[22] Mai, Xiaodong, Jiangke Cheng, and Shengnan Wang. "RETRACTED ARTICLE: Research on semi supervised K-means clustering algorithm in data mining." *Cluster Computing* 22, no. Suppl 2 (2019): 3513-3520.

[23] Rodriguez, Mayra Z., Cesar H. Comin, Dalcimar Casanova, Odemir M. Bruno, Diego R. Amancio, Luciano da F. Costa, and Francisco A. Rodrigues. "Clustering algorithms: A comparative approach." *PloS one* 14, no. 1 (2019): e0210236.

[24] Monalisa, Siti, and Fitra Kurnia. "Analysis of DBSCAN and K-means algorithm for evaluating outlier on RFM model of customer behaviour." *Telkomnika (Telecommunication Computing Electronics and Control)* 17, no. 1 (2019): 110-117.

[25] Shahriar, Nafi, SM Akib Al Faisal, Md Masfakuzzaman Pinjor, Md Al Sharif Zobayer Rafi, and Atiquer Rahman Sarkar. "Comparative Performance Analysis of K-Means and DBSCAN Clustering algorithms on various platforms." In *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, pp. 1-6. IEEE, 2019.

[26] Aggarwal, Deepshikha, and Deepti Sharma. "Application of clustering for student result analysis." *Int J Recent Technol Eng* 7, no. 6 (2019): 50-53.

[27] Jansen, Aren, Manoj Plakal, Ratheet Pandya, Daniel PW Ellis, Shawn Hershey, Jiayang Liu, R. Channing Moore, and Rif A. Saurous. "Unsupervised learning of semantic audio representations." In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 126-130. IEEE, 2018.

[28] Deepajothi.S, Dr.Juliana " Survey of Clustering Algorithm of Weka Tool on Labor Dataset" International Journal of Applied Engineering Research vol. 14, no. 5, pp. 90–95, 2019