__Original Research Article__

Classification of Cotton Leaf Diseases Using AlexNet and Machine Learning Models

ABSTRACT

Computer vision has been demonstrated as state-of-the-art technology in precision agriculture in recent years. In this paper, an Alex net model was implemented to identify and classify cotton leaf diseases. Cotton Dataset consists of 2275 images, in which 1952 images were used for training and 324 images were used for validation. Five convolutional layers of the AlexNet deep learning technique is applied for features extraction from raw data. They were remaining three fully connected layers of AlexNet and machine learning classification algorithms such as Ada Boost Classifier (ABC), Decision Tree Classifier (DTC), Gradient Boosting Classifier (GBC). K Nearest Neighbor (KNN), Logistic Regression (LR), Random Forest Classifier (RFC), and Support Vector Classifier (SVC) are used for classification. Three fully connected layers of Alex Net provided the best performance model with a 94.92% F1_score at the training time of about 51min.

Keywords: Cotton Disease Detection, Machine Learning, Deep Learning, F1_score.

1. INTRODUCTION

Computer vision has become a novel technology in various fields of applications such as medicine, machine vision. Computer vision performs image capturing, imaging processing, image analyzing, image classification, image reorganization, and named few advancements in deep learning techniques have led to automating the many computer vision tasks.

Cotton is one of the world's foremost and economy-driven crops for all agricultural-based countries. The reduction in cotton yield led to high economic loss to the farmers. Smart farming is vital to conduct disease incidence at a low level, good management strategies and taking preventive measures at the right time to reduce chemical usage and to increase production. Monitoring the crop during all stages of plant growth requires expert knowledge in the domain and extensive laborious work.

Among all deep learning techniques, convolution neural networks are a commonly employed method in image-based data applications. CNN offers feature extraction significantly easier with minimal human supervision and field knowledge than machine learning algorithms. The effectiveness of machine learning algorithms highly depends on the integrity of the input data representation. If the construction of features from raw data is poor, the machine learning algorithms may provide incorrect discrimination between data classes.

Hence, the present study was undertaken to predict the optimal model from various models, namely AlexNet and machine learning models, to classify cotton diseases based on the extracted image features.

2 MATERIAL AND METHODS

This section discusses data sets and hardware configuration details.

2.1 Datasets

An open-access cotton disease dataset [1] is used for training and validating the AlexNet model. The number of images in each training set and validating set contain 1951 and 324, respectively. The detailed information about the dataset, such as classes and labelling, is given in Tables 1 and 2. This dataset contains four distinct categories of images: fresh cotton leaves, fresh cotton plants, diseases cotton leaves, and diseased cotton plants, which are imbalanced datasets. Four classes in the training dataset don't have an approximately equal proportion.

Table 1. Imbalanced cotton train dataset

Type of Dataset	Category	No. of Images	Percentage of classes (%)
Train data	Diseased cotton leaf	288	14.76
	Diseased cotton plant	815	41.77
	Fresh cotton leaf	427	21.88
	Fresh cotton plant	421	21.57
	Total No. of training images	**1951**	

Table 2. imbalanced cotton validation dataset

Type of Dataset	Category	No. of Images	Percentage of (%) classes
Validation	Diseased cotton leaf	55	16.97
	Diseased cotton plant	101	31.17
	Fresh cotton leaf	80	24.69
	Fresh cotton plant	88	27.16
	Total No. of training images	**324**	

The sample images of four classes have shown in below figure1. From figure-1, Diseased cotton leaf.

(a) Diseased Cotton Leaf (b) Diseased Cotton Plant (c) Fresh Cotton Leaf (d) Fresh Cotton Plant

Figure 1. Sample images of cotton leaves and plants

2.2 Hardware and Software Specifications

All experiments are performed on a powerful machine, having the specifications that are summarized in Table 3.

Table 3 Hardware and software specifications

Hardware & Software	Characteristics
Memory (RAM)	16GB
Processor	Intel(R) Core (TM) i7-10875H CPU @ 2.30GHz 2.30 GHz
Graphics (GPU)	NVIDIA GeForce RTX 2070- 8GB
Operating system	Windows 10, 64 bits
Integrated Development Environment (IDE)	Jupyter Notebook

2.3 METHODOLOGY

The Research framework has three phases and is shown in figure 2. The first phase has to preprocess. The second phase has a feature engineering process that includes the CNN training techniques, namely AlexNet. The third phase had classification algorithms.

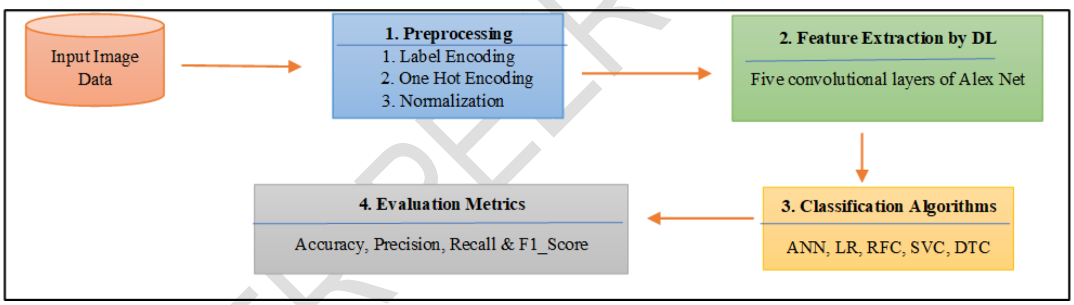


Fig.2 Framework for Disease Classification

2.3.1 Preprocessing Phase

The first phase is preprocessing of input image data with a size (h, w, c) of 227,227,3 that has been done in a sequence of operations. The input dataset classes are labelled through label encoding then apply a one-hot encoding technique. Here, class labels, namely disease cotton leaf, disease cotton plant, fresh cotton leaf and the fresh cotton plant, are text data. The system can’t understand the text data. So, we need to convert this kind of categorical text data into model-understandable numerical data with label encoding. The Lael encoding method will assign numbers between 0 and n-1, where n is a number of class labels (n=4) based on alphabetical order. Besides, one hot Encoding is another technique to treat categorical variables. This creates additional attributes based on the unique value in the categorical variable [2] [3]. Then the pixel values of images are normalized between 0 and 1.

2.3.2 Feature Extraction Phase

AlexNet is a deep CNN structure proposed by Krizhevsky and Sutskever [4]. AlexNet of CNN has two parts. And the first part is feature extraction, where to extract the features from the images and the second part is the classification part. Hyperparameters, namely optimizers like Adam and SGD (Stochastic Gradient Descent) [5], batch size like 32, 64,128, momentum [6] is 0.9, learning rate about 0.001 and dropout at 0.5 [7] were applied during training the model.

Table 4 DL Model parameters.

DL Model	Optimizers	BS	M	LR	DO
AlexNet	SGD & Adam	32, 64, 128	0.9	1e-3	0.5

Abbreviations: DL Model, Deep Learning Model; BS, Batch size; M, Momentum; LR, Learning Rate; DO, Dropout.

Table 5 Structure of DL Models.

Name of the DL Model	No. of Layers	Model Description
AlexNet	8	5 CL + 3 FCL

Abbreviations: CL, convolutional layer; FCL, fully connected layer

The number of features extracted from training images is 9216 by five convolutional layers of AlexNet, as shown in figure 3. At the end of the fifth convolutional layer, trainable parameters were obtained, non-trainable parameters were 3,748,416, 1216 respectively and 9216 features were extracted, and the extracted features fed to the remaining three fully connected layers of AlexNet for the classification. Among three fully connected layers, the first and second layers are hidden layers with 4096 nodes, and the third fully connected layer is the output layer with four classes, as shown in figure 4.

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 55, 55, 96)        34944
_________________________________________________________________
activation (Activation)      (None, 55, 55, 96)        0
_________________________________________________________________
batch_normalization (BatchNo (None, 55, 55, 96)        384
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 27, 27, 96)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 27, 27, 256)       614656
_________________________________________________________________
activation_1 (Activation)    (None, 27, 27, 256)       0
_________________________________________________________________
batch_normalization_1 (Batch (None, 27, 27, 256)       1024
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 256)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 13, 13, 384)       885120
_________________________________________________________________
activation_2 (Activation)    (None, 13, 13, 384)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 13, 13, 384)       1327488
_________________________________________________________________
activation_3 (Activation)    (None, 13, 13, 384)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 13, 13, 256)       884992
_________________________________________________________________
activation_4 (Activation)    (None, 13, 13, 256)       0
_________________________________________________________________
batch_normalization_2 (Batch (None, 13, 13, 256)       1024
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 9216)              0
=================================================================
Total params: 3,749,632
Trainable params: 3,748,416
Non-trainable params: 1,216
```

Fig 3 Structure of a 5-Conv layer of AlexNet

Layer (type)	Output Shape	Param #
conv2d_input (InputLayer)	[(None, 227, 227, 3)]	0
conv2d (Conv2D)	(None, 55, 55, 96)	34944
activation (Activation)	(None, 55, 55, 96)	0
batch_normalization (BatchNo	(None, 55, 55, 96)	384
max_pooling2d (MaxPooling2D)	(None, 27, 27, 96)	0
conv2d_1 (Conv2D)	(None, 27, 27, 256)	614656
activation_1 (Activation)	(None, 27, 27, 256)	0
batch_normalization_1 (Batch	(None, 27, 27, 256)	1024
max_pooling2d_1 (MaxPooling2	(None, 13, 13, 256)	0
conv2d_2 (Conv2D)	(None, 13, 13, 384)	885120
activation_2 (Activation)	(None, 13, 13, 384)	0
conv2d_3 (Conv2D)	(None, 13, 13, 384)	1327488
activation_3 (Activation)	(None, 13, 13, 384)	0
conv2d_4 (Conv2D)	(None, 13, 13, 256)	884992
activation_4 (Activation)	(None, 13, 13, 256)	0
batch_normalization_2 (Batch	(None, 13, 13, 256)	1024
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 256)	0
flatten (Flatten)	(None, 9216)	0
flatten_1 (Flatten)	(None, 9216)	0
dense (Dense)	(None, 4096)	37752832
dense_1 (Dense)	(None, 4096)	16781312
dropout (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 4)	16388

Total params: 58,300,164
Trainable params: 58,298,948
Non-trainable params: 1,216

Fig 4: Structure of 8 layers (5-conv+3-FC) layer of AlexNet

2.3.3 Classification phase

Features are extracted from the previous stage; those features feed the remaining three fully connected layers of AlexNet for training. Eight layers of AlexNet model structure have shown in figure 4. Traditional machine learning algorithms, namely, ABC [8,9], is ensemble classifier that combines weak classifiers and forms a robust classifier. DTC [10,11] is a supervised algorithm used for both regression and classification problems. GBC [12,13] is a supervised learning algorithm where strong predictors are built with the help of weak predictors. KNN [14,15] is also used for regression and classification, which works well on multiclass problems. LR [16,17] is computationally efficient, and scaling is not required for this model. RFC [18,19], Can be used for regression and classification problems. It can Solve overfitting a problem. SVC [20,21] can handle linear and non-linear data, and there is less probability of overfitting.

3. RESULTS AND DISCUSSION

In this experiment, while extracting the features by deep learning models, apply different hyperparameters like batch sizes, learning rate, momentum, decay, number of epochs.

3.1 Performance Measure

Evaluated the performance of classification models through a confusion matrix from figure5. Evaluation metrics are Accuracy, Precision, Recall (Sensitivity), Specificity, F1_Score. A combination of precession and recall is the F1 score, which is used when the dataset belongs imbalanced. The given dataset is imbalanced so, we need to choose the optimal model based on macro average f1_score. True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) parameters of the confusion matrix were used to calculate the metrics [22,23] for 'k' classes.

		PREDICTED LABELS				
		L_1	L_2	…	L_K	
ACTUAL LABELS	L_1	**TP_1**				TPR_1
	L_2		**TP_2**			TPR_2
	L_3	…	…	**…**	…	…
	L_K				**TP_K**	TPR_K
		PPV_1	PPC_2	…	PPV_K	

Fig 5. The confusion matrix for the multiclass classification

3.2 Experimental Results

3.2.1 AlexNet Results

Five convolutional layers of the AlexNet model are responsible for extracting the features. Those features are fed to the remaining three fully connected layers of AlexNet and machine learning algorithms for classifying the classes. Those results are shown in table 6 and table 7. The best results are 94.72% of a macro average of F1_score found from table 7 at hyperparameters batch size 64, learning rate 0.001, decay value is 1e-6, momentum is 0.9 and number of epochs are 50 for getting these optimal results to train the AlexNet model for approximately 51 minutes.

Table 6. Results of AlexNet on No. of features: 9216

Results of **SGD** Optimizer															
BS	**LR**	**D**	**M**	**E**	**Train Score (%)**	**Trian Loss**	**Test Score (%)**	**Test loss**	**Precision**		**Recall**		**F1_Score**		**Computation time (Sec)**
									M.A	**W. A**	**M.A**	**W. A**	**M.A**	**W. A**	
32	0.001	1e-6	0.9	50	100.0	0.0006	91.35	0.31	92.66	91.72	91.35	91.36	91.88	91.40	3386
64	**0.001**	**1e-6**	**0.9**	**50**	**100.0**	**0.0003**	**94.75**	**0.18**	**95.02**	**94.95**	**95.02**	**94.75**	**94.92**	**94.74**	**3053**
128	0.001	1e-6	0.9	50	99.89	0.006	91.35	0.25	91.67	91.33	91.98	91.36	91.78	91.30	3074
Results of **Adam** Optimizer															
32	0.001	1e-6	-	50	85.90	0.91	70.68	2.74	79.61	77.36	71.02	70.68	70.26	68.73	3456
64	0.001	1e-6	-	50	76.11	1.32	60.49	3.30	65.97	64.67	61.69	60.49	59.96	58.66	3171
128	0.001	1e-6	-	50	83.90	0.95	62.65	3.0	73.57	73.09	63.50	62.65	61.54	60.56	3000

Abbreviations: BS, Batch size; LR, Learning Rate; D, Decay; M, Momentum; E, No. of Epochs; M. A, Macro Average; W. A, Weighted Average

Table 7: Results of classification models

Name of the Model	Train time (Sec)	Train Accuracy	Test Accuracy	Precession		Recall		F1_Score	
				Macro	Weighted	Macro	Weighted	Macro	Weighted
AlexNet	**3053**	**100.0**	**94.75**	**95.02**	**94.95**	**95.02**	**94.75**	**94.92**	**94.74**
ABC	46	74.21	69.75	73.41	71.31	69.90	69.75	71.27	70.18
DTC	08	100.0	65.43	66.12	66.16	65.78	65.43	65.70	65.56
GBC	894	100.0	84.87	87.35	85.97	84.47	84.88	85.39	84.85
KNN	03	80.31	72.84	80.62	77.67	68.67	72.84	69.55	71.50
LR	02	78.06	75.92	82.03	79.52	74.52	75.93	76.19	75.66
RFC	04	100.0	78.08	84.16	82.22	76.50	78.09	78.33	78.10
SVC	149	76.52	74.69	81.21	78.48	73.34	74.69	75.05	74.40

Training accuracy, validation accuracies and training loss, validation loss of the AlexNet model were shown in Figures 6 and 7, respectively. From figure 6, while epochs were increasing, accuracy was also increased for training the AlexNet model for fifty epochs taken training time about 3053 seconds and reached training accuracy about 100% and validation accuracy about 94.75% and macro f1_score approximately 94.92%. Training data is imbalanced data so, based on macro f1_score to choose the deployment model. From table 7, among all classification models, the AlexNet model has the best f1_score.

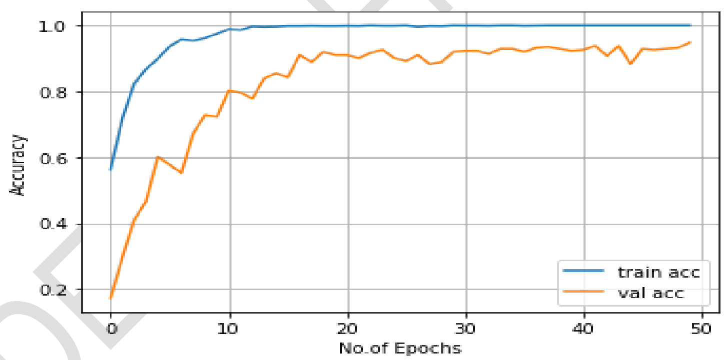


Fig 6 Train accuracy and validation accuracy

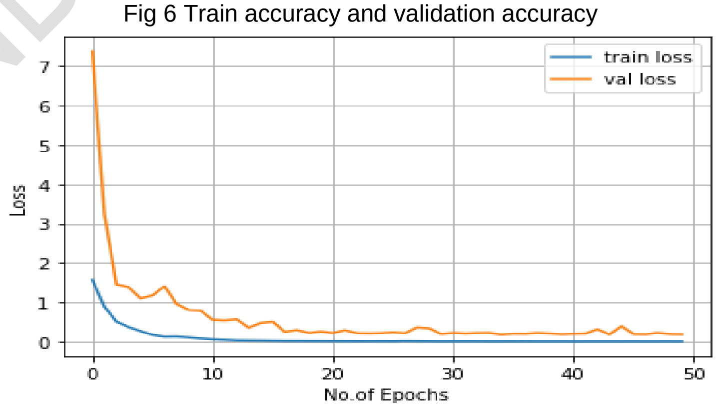


Fig 7 Train loss and validation loss

The confusion matrix of the AlexNet model is shown in Figure 8. From the confusion matrix, the diseased cotton leaf is correctly predicted 54 samples out of 55 samples. Disease cotton plant is perfectly expected 98 samples out of 101 samples. The fresh cotton leaf is perfectly 77 samples out of 80 samples. The fresh cotton plant is correctly predicted 78 samples out of 88 samples. The classification report of the AlexNet model has shown in figure 9. The comparison of classification models and the AlexNet model are shown in figure 10.

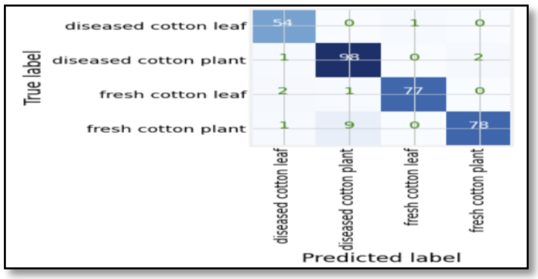
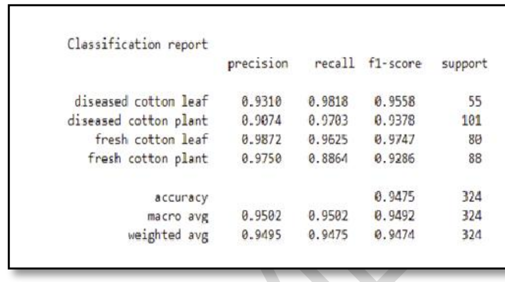


Fig. 8 Confusion matrix of AlexNet on validation data


Classification report

	precision	recall	f1-score	support
diseased cotton leaf	0.9310	0.9818	0.9558	55
diseased cotton plant	0.9074	0.9703	0.9378	101
fresh cotton leaf	0.9872	0.9625	0.9747	80
fresh cotton plant	0.9750	0.8864	0.9286	88
accuracy			0.9475	324
macro avg	0.9502	0.9502	0.9492	324
weighted avg	0.9495	0.9475	0.9474	324

Fig. 9 Classification report of AlexNet on validation data

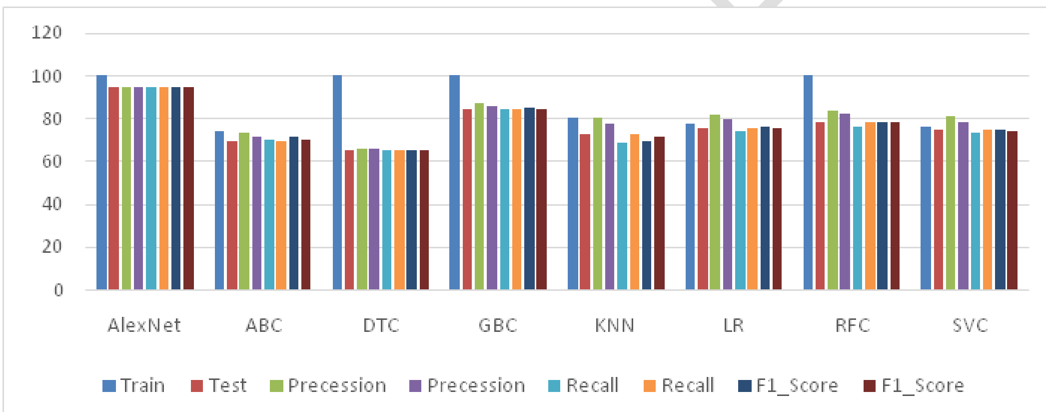


Fig. 10 Comparison of all models

4. CONCLUSION

Five convolutional layers of the AlexNet model is used to extract the features from images, and those features are feed to three fully connected layers of AlexNet and machine learning classification models for classifying the cotton leaf disease. Among all models, the AlexNet model has given the best results. Given dataset is imbalanced data, for imbalanced data, based on the macro F1_score, we choose the optimal deployment model. Among all classification models, the AlexNet model has given the best result of F1_score is 94.92%.

REFERENCES

[1] HTTPS://WWW.KAGGLE.COM/JANMEJAYBHOI/COTTON-DISEASE-DATASET.

[2] GU, B., & SUNG, Y. (2021). ENHANCED REINFORCEMENT LEARNING METHOD COMBINING ONE-HOT ENCODING-BASED VECTORS FOR CNN-BASED ALTERNATIVE HIGH-LEVEL DECISIONS. APPLIED SCIENCES, 11(3), 1291.

[3] YANG, X., HOU, L., ZHOU, Y., WANG, W., & YAN, J. (2021). DENSE LABEL ENCODING FOR BOUNDARY DISCONTINUITY FREE ROTATION DETECTION. IN PROCEEDINGS OF THE IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 15819-15829).

[4] KRIZHEVSKY A ET AL. (2012) IMAGENET CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS. IN: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS. 1097–1105.

[5] PU, S., OLSHEVSKY, A., & PASCHALIDIS, I. C. (2021). A SHARP ESTIMATE ON THE TRANSIENT TIME OF DISTRIBUTED STOCHASTIC GRADIENT DESCENT. IEEE TRANSACTIONS ON AUTOMATIC CONTROL.

[6] SUN, Y., LIU, Y., ZHOU, H., & HU, H. (2021). PLANT DISEASES IDENTIFICATION THROUGH A DISCOUNT MOMENTUM OPTIMIZER IN DEEP LEARNING. APPLIED SCIENCES, 11(20), 9468.

[7] PHAM, H., & LE, Q. V. (2021). AUTODROPOUT: LEARNING DROPOUT PATTERNS TO REGULARIZE DEEP NETWORKS. ARXIV PREPRINT ARXIV:2101.01761, 1(2), 3.

[8] WASHBURN, P. S. (2020). INVESTIGATION OF SEVERITY LEVEL OF DIABETIC RETINOPATHY USING ADABOOST CLASSIFIER ALGORITHM. MATERIALS TODAY: PROCEEDINGS, 33: 3037-3042.

[9] KUMAR, C. S., SHARMA, V. K., YADAV, A. K., & SINGH, A. (2021). PERCEPTION OF PLANT DISEASES IN COLOR IMAGES THROUGH ADABOOST. IN INNOVATIONS IN COMPUTATIONAL INTELLIGENCE AND COMPUTER VISION (PP. 506-511). SPRINGER, SINGAPORE.

[10] PRIYANKA, & KUMAR, D. (2020). DECISION TREE CLASSIFIER: A DETAILED SURVEY. INTERNATIONAL JOURNAL OF INFORMATION AND DECISION SCIENCES, 12(3): 246-269.

[11] KOGA, S., ZHOU, X., & DICKSON, D. W. (2021). MACHINE LEARNING-BASED DECISION TREE CLASSIFIER FOR THE DIAGNOSIS OF PROGRESSIVE SUPRANUCLEAR PALSY AND CORTICOBASAL DEGENERATION. NEUROPATHOLOGY AND APPLIED NEUROBIOLOGY.

[12] TANHA, J., ABDI, Y., SAMADI, N., RAZZAGHI, N., & ASADPOUR, M. (2020). BOOSTING METHODS FOR MULTICLASS IMBALANCED DATA CLASSIFICATION: AN EXPERIMENTAL REVIEW. JOURNAL OF BIG DATA, 7(1): 1-47.

[13] SHRIVASTAV, L. K., & JHA, S. K. (2021). A GRADIENT BOOSTING MACHINE LEARNING APPROACH IN MODELING THE IMPACT OF TEMPERATURE AND HUMIDITY ON THE TRANSMISSION RATE OF COVID-19 IN INDIA. APPLIED INTELLIGENCE, 51(5), 2727-2739.

[14] ABU ALFEILAT, H. A., HASSANAT, A. B., LASASSMEH, O., TARAWNEH, A. S., ALHASANAT, M. B., EYAL SALMAN, H. S., & PRASATH, V. S. (2019). EFFECTS OF DISTANCE MEASURE CHOICE ON K-NEAREST NEIGHBOR CLASSIFIER PERFORMANCE: A REVIEW. BIG DATA, 7(4): 221-248.

[15] PATIL, A., & LAD, K. (2021). CHILI PLANT LEAF DISEASE DETECTION USING SVM AND KNN CLASSIFICATION. IN RISING THREATS IN EXPERT APPLICATIONS AND SOLUTIONS (PP. 223-231). SPRINGER, SINGAPORE.

[16] SONG, X., LIU, X., LIU, F., & WANG, C. (2021). COMPARISON OF MACHINE LEARNING AND LOGISTIC REGRESSION MODELS IN PREDICTING ACUTE KIDNEY INJURY: A SYSTEMATIC REVIEW AND META-ANALYSIS. INTERNATIONAL JOURNAL OF MEDICAL INFORMATICS, 104484.

[17] XIAO, R., CUI, X., QIAO, H., ZHENG, X., & ZHANG, Y. (2021). EARLY DIAGNOSIS MODEL OF ALZHEIMER'S DISEASE BASED ON SPARSE LOGISTIC REGRESSION. MULTIMEDIA TOOLS AND APPLICATIONS, 80(3), 3969-3980.

[18] MORE, A. S., & RANA, D. P. (2017, OCTOBER). REVIEW OF RANDOM FOREST CLASSIFICATION TECHNIQUES TO RESOLVE DATA IMBALANCE. IN 2017 1ST INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND INFORMATION MANAGEMENT (ICISIM) (PP. 72-78). IEEE.

[19] SAHA, S., & AHSAN, S. M. M. (2021, FEBRUARY). RICE DISEASE DETECTION USING INTENSITY MOMENTS AND RANDOM FOREST. IN 2021 INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY FOR SUSTAINABLE DEVELOPMENT (ICICT4SD) (PP. 166-170). IEEE.

[20] GHADDAR, B., & NAOUM-SAWAYA, J. (2018). HIGH DIMENSIONAL DATA CLASSIFICATION AND FEATURE SELECTION USING SUPPORT VECTOR MACHINES. EUROPEAN JOURNAL OF OPERATIONAL RESEARCH, 265(3): 993-1004.

[21] RAGHAVENDRA, Y. (2021). MULTIVARIANT DISEASE DETECTION FROM DIFFERENT PLANT LEAVES AND CLASSIFICATION USING MULTICLASS SUPPORT VECTOR MACHINE. TURKISH JOURNAL OF COMPUTER AND MATHEMATICS EDUCATION (TURCOMAT), 12(13), 546-556.

[22] SOKOLOVA, M., & LAPALME, G. (2009). A SYSTEMATIC ANALYSIS OF PERFORMANCE MEASURES FOR CLASSIFICATION TASKS. INFORMATION PROCESSING & MANAGEMENT, 45(4): 427-437. [23] GRANDINI, M., BAGLI, E., & VISANI, G. (2020). METRICS FOR MULTI-CLASS CLASSIFICATION: AN OVERVIEW. ARXIV PREPRINT ARXIV:2008.05756.