

Big Data and Machine Learning Based Early Chronic Kidney Disease Prediction

Asra Fatima¹, Shireen Fatima², Ayesha Kiran³

¹Assistant Professor, Department of Computer Science and Engineering Faculty of Engineering and Technology, KBN University, Gulbarga, Karnataka, India. asra@kbn.university

²Assistant Professor, Department of Computer Science and Engineering Faculty of Engineering and Technology, KBN University, Gulbarga, Karnataka, India. shireen1@kbn.university

³Assistant Professor, Department of Computer Science and Engineering Faculty of Engineering and Technology, KBN University, Gulbarga, Karnataka, India. kiran@kbn.university

ABSTRACT

A chronic kidney disease, sometimes called a chronic renal disease, is characterized by a gradual decline in kidney purpose or abnormal kidney purpose which continues for months or even years. Patients with a domestic past of chronic kidney disease (CKD), high BP, or other kidney-related conditions are often the first to have chronic kidney disease (CKD) identified during screenings. Consequently, effective illness prevention and therapy rely on early prediction. Methods from the field of machine learning, including XGBoost, KNN, Decision Tree, and Random Forest, are being considered for use in this CKD project. The final product uses the fewest characteristics possible to determine whether the patient has chronic kidney disease (CKD).

Keywords- Random forest, Decision tree, XGBoot, KNN

I. INTRODUCTION

Current medical consensus is that chronic kidney disease (CKD) stances a important risk to public fitness. Laboratory tests can detect chronic kidney disease on a regular basis, and there are treatments that can stop the disease from progressing, slow it down, lessen danger of cardiovascular disease, its complications, increase subsistence, improve eminence of life.

Here, we look at the pros and cons of using ML for early CKD detection, as well as its practicality. Our goal is to create a machine learning model that uses split ratio, optimum parameters, data imputation, and data scaling approaches to assess classifiers' performance in classification tasks. The objective is to find a way to use ML techniques like DT, k-nearest to diagnose CKD efficiently. We use iterative imputation to manage lost digits, offer a new consecutive data scaling method. To find the most relevant features, we use Boruta feature selection, and then we modify the hyperparameters using grid-search CV. We compare our planned work's testing accuracy to those of existing research to see how well it holds up.

II. RELATED WORK

J. Snegha, integrated a number of data mining tools, including a Backpropagation Neural Network and the Random Forest method, into their suggested system. Because it employs a supervised learning network known as a feedforward neural network, the Back Propagation technique outperforms the other approach in this comparison.[1].

Mohamed Elhoseny, K. Shankar & J. Uthayakumar , detailed a CKD system that employs ACO-based density-based feature selection. When choosing features, the system employs wrapper techniques.[2].

Baisakhi Chakraborty, usage of machine learning algorithms including K-Nearest Neighbor, Logistic Regression, Decision Tree, Random Forest Multi-Layer Perceptron Algorithm for the suggested construction of a system to detect chronic kidney disease. After applying them, we compare their performance to the outcomes of our recall, precision, and accuracy tests. And lastly, this system's implementation is based on Random Forest.[3].

International Conference on ECCE, developed a method for illness prediction using Boosting Classifiers, Ant-Miner, J48 Decision Tree. Both assessing efficacy of boosting algorithms in CKD detection and drawing rules to show correlations between CKD characteristics are the goals of this work. Based on the findings of the experiments, AdaBoost performed somewhat worse than LogitBoost.[4].

Siddeshwar Tekale, detailed a machine learning system which use support vector machine (SVM) decision-tree approaches. After evaluating two methods, we found that SVM produces the best outcome. So that physicians may examine patients in less time, its prediction procedure is less time-consuming. [5].

III. OBJECTIVES

1. Use the given information to make a prediction about the likelihood of chronic renal disease in a given person.
2. In order to create and verify a model that may predict the onset of chronic renal disease. Predicting and classifying illnesses is a common use case for machine learning algorithms in medicine.

IV. EXISTING SYSTEM

We analyzed the current system using three machine learning classifiers—Logistic Regression (LR), Decision Tree (DT), and SVM—upgraded the model's performance using the bagging ensemble approach. Clusters in the chronic renal illness dataset were used to train machine learning classifiers. We next use nonlinear characteristics and categories to construct the Kidney Disease Collection. The decision tree outperforms the others and has a 95% success rate. One more thing: bagging ensemble method gets us to highest possible correctness of 97%.

V. PROPOSED SYSTEM

Machine learning algorithms including XGBoost, Decision Tree, Random Forest, and K-Nearest Neighbor are to be used in proposed CKD prediction system. After that, we compare their efficacy to the outcomes of the accuracy, precision, and recall tests.

A. ADVANTAGES OF PROPOSED SYSTEM

1. Predicts, prevents and detects the disease
2. Needs less time
3. Accuracy is more

B. ARCHITECTURE

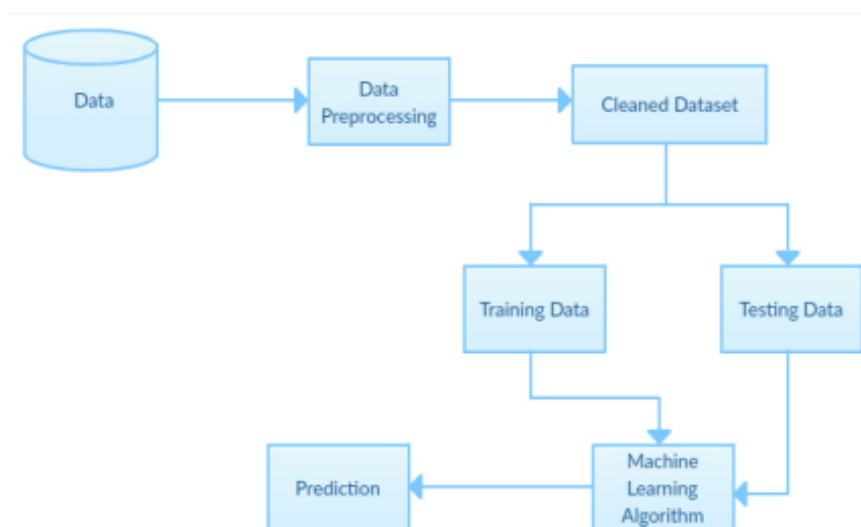


Fig-1: System Architecture

VI. METHODOLOGY

A. Decision Tree (DT):

The design of the decision tree classifier is reminiscent of a tree or a flowchart. A parent node, branches, and child nodes make it up. Subdivisions represent the result of each check on each node, whereas inner nodes include the features. Due to its ability to function without a large amount of field data or restrictions, decision trees are among the most popular classifiers used for categorization determination.

B. K-Nearest Neighbor

This supervised ML technique is popular because to its simplicity. It uses a distance measure like the Manhattan, Minkowski, or Euclidean to forecast class of an remark by looking at classes of its k closest neighbors. Here are the formulas for various distance measurements:

$$d_{Euclidean}(x_i, x_j) = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2}, \quad (6)$$

$$d_{Minkowski}(x_i, x_j) = \left(\sum_{k=1}^d |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}}, \quad (7)$$

$$d_{Manhattan}(x_i, x_j) = \sum_{k=1}^d |x_{ik} - x_{jk}|. \quad (8)$$

Here, x_{ik} and x_{ij} stand for the k th and d -th characteristics of x_i , x_j , respectively, in a space of d dimensions. By these distance measurements, it finds a data point's k closest neighbors and uses the class that is most prevalent among them to control data point's class. Algorithm's ease of use make it suitable for a wide range of categorization jobs.

Algorithm provides the pseudocode that outlines step-by-step approach, crucial hyperparameter selections for building, Gaussian NB model in this study. Model construction makes use of hyperparameters including training data, leaf size, parameter, weight function, method, number of neighbors, distance metric. Taking into account the majority class among a test instance's k closest neighbors, the method predicts its class label. The method uses a majority vote procedure to determine the predicted class label after computing detachments among test instance, training examples, picking k closest neighbors, so on.

The k-nearest neighbors pseudocode.	
Input:	
Training dataset: $X_{train} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$	
Test instance: X_{test}	
algorithm: ['auto', 'ball_tree', 'kd_tree', 'brute']	
leaf_size: [20, 30, 40]	
metric: ['euclidean', 'manhattan', 'minkowski']	
n_neighbors: [3, 5, 7]	
p: [3, 4]	
weights: ['uniform', 'distance']	
Output:	
Predicted class label for X_{test} : $\hat{y}$	
1: procedure KNN(X_{train} , X_{test} , algorithm, leaf_size, metric, n_neighbors, p, weights)	
2: distances $\leftarrow$ Empty list	
3: for each (x_i, y_i) in X_{train} do	
4: distance $\leftarrow$ Calculate the distance between x_i and X_{test} using metric	
5: Add (distance, y_i) to distances	
6: end for	
7: Sort distances in ascending order based on distance	
8: k_nearest $\leftarrow$ First n_neighbors elements from distances	
9: labels $\leftarrow$ Extract labels from k_nearest	
10: if weights is 'distance', then	
11: Compute weights based on the distance for labels	
12: end if	
13: $\hat{y} \leftarrow$ Perform a weighted majority vote of labels using weights	
14: return $\hat{y}$	
15: end procedure	

Fig-2: Pseudo Code

VII. IMPLEMENTATION

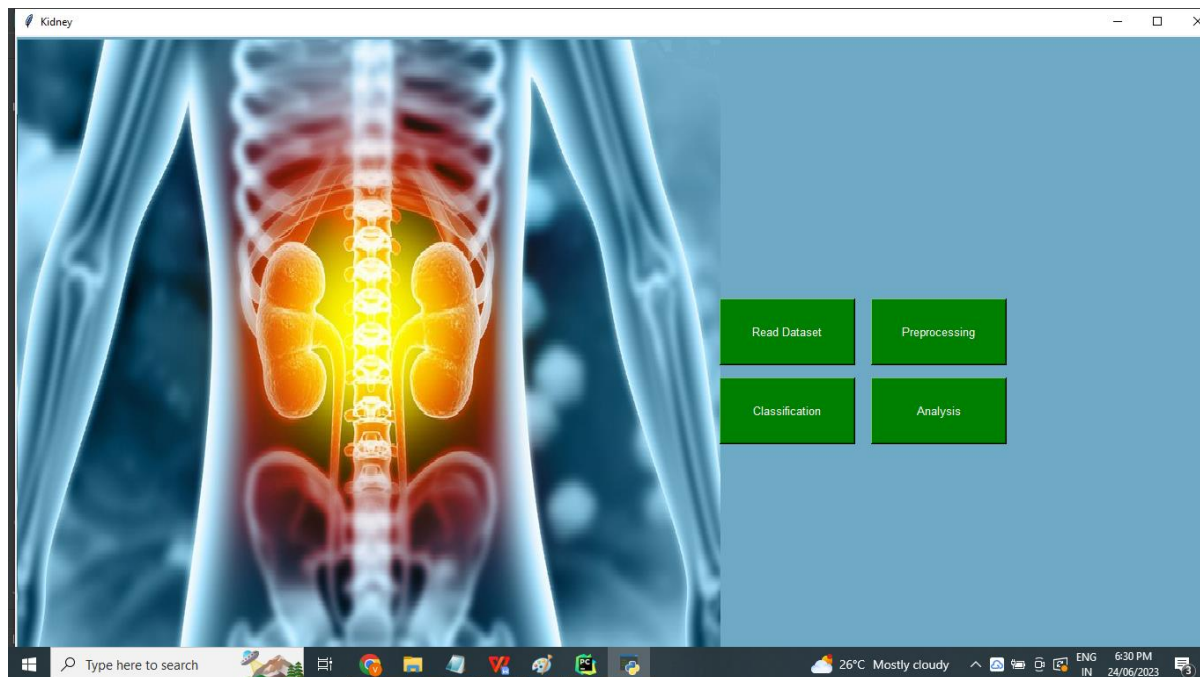


Fig-3: Home Page

id	age	bp	sg	al	Su	rbc	pc	pcc
399	58	80	1.025	0	0	normal	normal	notpresent
398	17	60	1.025	0	0	normal	normal	notpresent
397	12	80	1.02	0	0	normal	normal	notpresent
396	42	70	1.025	0	0	normal	normal	notpresent
395	55	80	1.02	0	0	normal	normal	notpresent
394	50	80	1.02	0	0	normal	normal	notpresent
393	43	60	1.025	0	0	normal	normal	notpresent
392	57	80	1.02	0	0	normal	normal	notpresent
391	36	80	1.025	0	0	normal	normal	notpresent
390	52	80	1.025	0	0	normal	normal	notpresent
389	41	80	1.025	0	0	normal	normal	notpresent
388	51	80	1.02	0	0	normal	normal	notpresent
387	15	80	1.025	0	0	normal	normal	notpresent
386	46	70	1.025	0	0	normal	normal	notpresent
385	63	70	1.02	0	0	normal	normal	notpresent
384	57	60	1.02	0	0	normal	normal	notpresent
383	80	80	1.025	0	0	normal	normal	notpresent
382	48	80	1.025	0	0	normal	normal	notpresent
381	71	70	1.025	0	0	normal	normal	notpresent
380	59	60	1.02	0	0	normal	normal	notpresent
379	62	80	1.025	0	0	normal	normal	notpresent
378	71	60	1.025	0	0	normal	normal	notpresent
377	64	70	1.02	0	0	normal	normal	notpresent
376	58	70	1.025	0	0	normal	normal	notpresent
375	70	80	1.02	0	0	normal	normal	notpresent
374	79	80	1.025	0	0	normal	normal	notpresent
373	61	70	1.025	0	0	normal	normal	notpresent
372	72	60	1.02	0	0	normal	normal	notpresent
371	28	60	1.025	0	0	normal	normal	notpresent
370	69	70	1.02	0	0	normal	normal	notpresent
369	75	70	1.02	0	0	normal	normal	notpresent
368	30	80	1.025	0	0	normal	normal	notpresent
367	68	60	1.025	0	0	normal	normal	notpresent
366	60	80	1.025	0	0	normal	normal	notpresent
365	24	80	1.02	0	0	normal	normal	notpresent
364	73	80	1.025	0	0	normal	normal	notpresent

Fig-4: Dataset

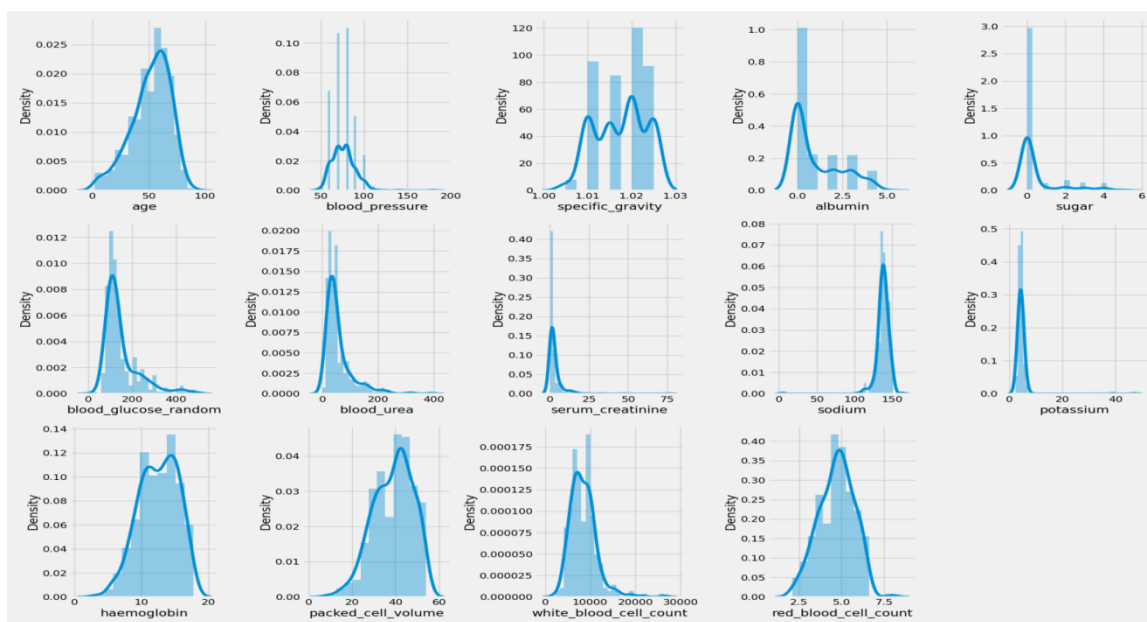


Fig-5: Graphical Dataset

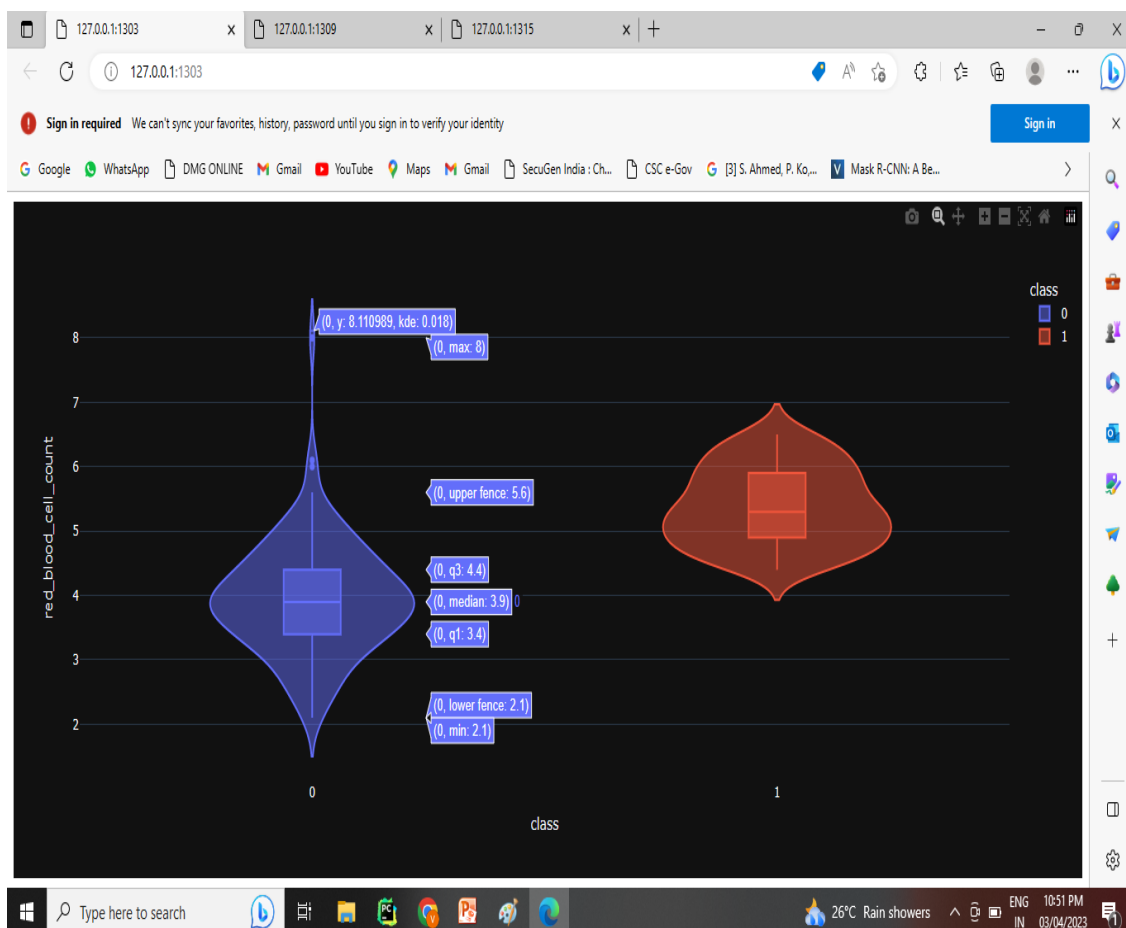


Fig-6: Classification

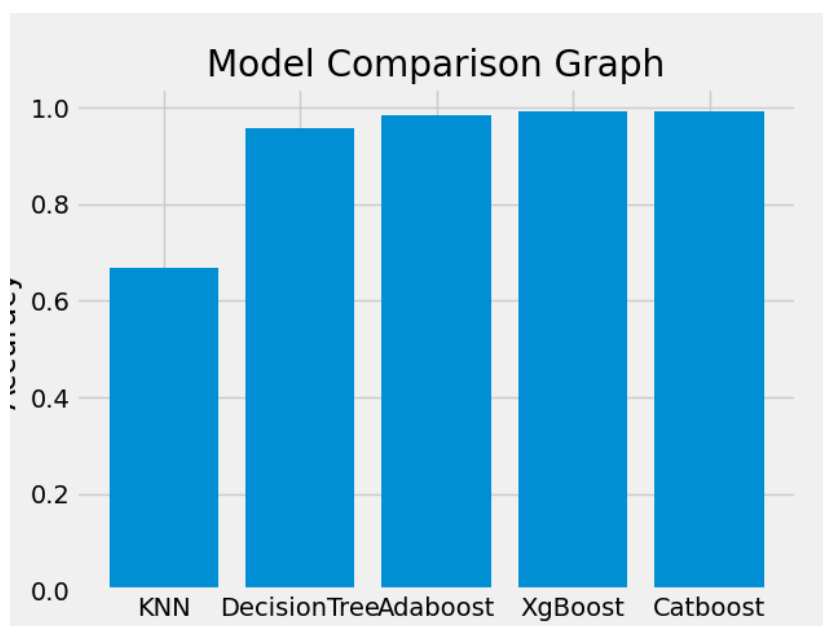


Fig-7: Model Comparison

VIII. CONCLUSION

Analysis of lasting kidney disease in humans is focus of this program. In this study, we compared the accuracy of 14 patient-related characteristics of chronic kidney disease (CKD) using several machine learning methods, like decision tree, random forest. One benefit of this method is the time savings it provides throughout the forecast process. Medical professionals will be able to diagnose more patients in less time and begin treating CKD patients earlier with this technology.

REFERENCES

1. J. Snegha, "Chronic Kidney Disease Prediction using Data Mining", International Conference on Emerging Trends, 2020.
2. Intelligent Diagnostic Prediction and Classification System for Chronic Kidney Disease, Mohamed Elhoseny, K. Shankar & J. Uthayakumar, *Scientific Reports* volume 9, Article number: 9583 (2019)
3. Baisakhi Chakraborty, "Development of Chronic Kidney Disease Prediction Using Machine Learning", International Conference on Intelligent Data Communication Technologies, 2019.
4. Rule Induction and Prediction of Chronic Kidney Disease Using Boosting Classifiers, Ant-Miner and J48 Decision Tree, February 2019, Conference: 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), At: Cox'sBazar, Bangladesh, Bangladesh
5. Siddeshwar Tekale, "Prediction of Chronic Kidney Disease Using Machine Learning, International Journal of Advanced Research in Computer and Communication Engineering, 2018.
6. Kai-Cheng Hu, "Multiple Pheromone table based on Ant Colony Optimization for Clustering", Hindawi, Research article, 2015.
7. Guneet Kaur, "Predict Chronic Kidney Disease using Data Mining in Hadoop, International Conference on Inventive Computing and Informatics, 2017.
8. Siddeshwar Tekale, "Prediction of Chronic Kidney Disease Using Machine Learning, International Journal of Advanced Research in Computer and Communication Engineering, 2018.
9. Luyckx, V.A.; Al-Aly, Z.; Bello, A.K.; Bellorin-Font, E.; Carlini, R.G.; Fabian, J.; Garcia-Garcia, G.; Iyengar, A.; Sekkarie, M.; Van Biesen, W.; et al. Sustainable development goals relevant to kidney health: An update on progress. *Nat. Rev. Nephrol.* **2021**, *17*, 15–32. [[Google Scholar](#)] [[CrossRef](#)]

10. Hoste, E.A.; Kellum, J.A.; Selby, N.M.; Zarbock, A.; Palevsky, P.M.; Bagshaw, S.M.; Goldstein, S.L.; Cerdá, J.; Chawla, L.S. Global epidemiology and outcomes of acute kidney injury. *Nat. Rev. Nephrol.* **2018**, *14*, 607–625. [[Google Scholar](#)] [[CrossRef](#)]
11. Dr. Sharanabasappa Gandage, Poornachandra Swami, Rohit, & Shankar Reddy Bengire. (2023). Virtual Paint. *Journal of Scientific Research and Technology*, 1(2), 25–38.
12. Dr. Rekha J Patil, Indira Mulage, & Nishant Patil. (2023). Smart Agriculture Using IoT and Machine Learning. *Journal of Scientific Research and Technology*, 1(3), 47–59.
13. Shubam Sharma, & Prof. Vinod Sharma. (2023). Comparison of machine learning techniques in the diagnosis of erythematous squamous disease. *Journal of Scientific Research and Technology*, 1(4), 1–9.
14. Deeksha Nargotra, & Prof. Vinod Sharma. (2023). Grape Leaf Disease Detection Using Deep Learning. *Journal of Scientific Research and Technology*, 1(5), 128–139.
15. Hassina Kouser, & Prof. Vibhakar Mansotra. (2023). Deep Learning-Based Segmentation Of Brain Tumor. *Journal of Scientific Research and Technology*, 1(5), 78–91.
16. Sabina Anjum, & Asra Fatima. (2023). Predictive Analytics For FIFA Player Prices: An ML Approach. *Journal of Scientific Research and Technology*, 1(6), 204–212.
17. Antariksh Sharma, Prof. Vibhakar Mansotra, & Kuljeet Singh. (2023). Detection of Mirai Botnet Attacks on IoT devices Using Deep Learning. *Journal of Scientific Research and Technology*, 1(6), 174–187.
18. Dr. Megha Rani Raigonda, & Rajeshwari Hatti. (2024). Haze Removal Of Underwater Images Using Fusion Technique. *Journal of Scientific Research and Technology*, 2(1), 12–16. <https://doi.org/10.61808/jsrt80>
19. Dr. Megha Rani Raigonda, & Shweta. (2024). Signature Verification System Using SSIM In Image Processing. *Journal of Scientific Research and Technology*, 2(1), 5–11. <https://doi.org/10.61808/jsrt79>
20. Shri Udayshankar B, Veeraj R Singh, Sampras P, & Aryan Dhage. (2023). Fake Job Post Prediction Using Data Mining. *Journal of Scientific Research and Technology*, 1(2), 39–47.
21. Gaurav Prajapati, Avinash, Lav Kumar, & Smt. Rekha S Patil. (2023). Road Accident Prediction Using Machine Learning. *Journal of Scientific Research and Technology*, 1(2), 48–59.
22. Dr. Rekha Patil, Vidya Kumar Katrabad, Mahantappa, & Sunil Kumar. (2023). Image Classification Using CNN Model Based on Deep Learning. *Journal of Scientific Research and Technology*, 1(2), 60–71.
23. Ambresh Bhadrashetty, & Surekha Patil. (2024). Movie Success and Rating Prediction Using Data Mining. *Journal of Scientific Research and Technology*, 2(1), 1–4. <https://doi.org/10.61808/jsrt78>
24. Dr. Megha Rani Raigonda, & Shweta. (2024). Signature Verification System Using SSIM In Image Processing. *Journal of Scientific Research and Technology*, 2(1), 5–11. <https://doi.org/10.61808/jsrt79>
25. Priyanka Kulkarni, & Dr. Swaroopa Shastri. (2024). Rice Leaf Diseases Detection Using Machine Learning. *Journal of Scientific Research and Technology*, 2(1), 17–22. <https://doi.org/10.61808/jsrt81>