International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Human Posture Recognition by Distribution-Aware Coordinate Representation and Machine Learning

Indrajit De[1], Lekha Rani[2], Rajat Bhardwaj[3], Ambuj Kumar Agarwal[*4], Raj Gaurang Tiwari[5]

Submitted: 24/01/2023 **Revised:** 15/03/2023 **Accepted:** 09/04/2023

Abstract: There has been a lot of research put into the statistical study of human behavior and movement. The ability to infer behavior from a single picture or a series of photos is a hot research area right now. Human Posture Recognition is a significant breakthrough in the direction of behavior comprehension since it may be used to identify actions taking place in a picture. Human posture estimate from the video is crucial for a wide range of uses, including the measurement of workouts, the identification of signs, and the manipulation of whole bodies via gestures. It may serve as the foundation for many dance, fitness, and yoga practices. It may also make augmented reality possible, where digital data is superimposed on the actual environment.The purpose of this study is to investigate and evaluate the viability of using Machine Learning to categorize human body position alongside a wide variety of complicated physical activities. Different basic, boosting, and ensemble machine learning methods are used in this study to categorize human posture based on the positions of individual body components (Distribution-aware coordinate representation). This study's dataset has 10 distinct physical positions that may be used to categorize 5 distinct workouts. These routines include variations on the push-up, pull-up, sit-up, jumping Jack, and squat. The final states of each exercise (the "up" and "down" postures for push-ups, for example) have been represented by two distinct classes. The strong predictions offered by the ensemble techniques were the result of the aggregation of the efforts of many different learners, making them more flexible.

Keywords: *Human Posture, Supervised Machine Learning, Classification, feature extraction*

1. Introduction

There has been a dramatic change in the focus of computer applications from traditional data processing to machine learning in recent years as a result of the widespread availability and accessibility of huge volumes of data generated by sensors and the internet[1], [2]. Machine learning is an idea that has expanded widely because it provides concrete evidence that machines can learn and improve[3]. By hosting conferences, seminars, group discussions, experiments, and actual implementations, western nations have shown a strong interest in the fields of machine learning, computer vision, and pattern recognition[4]. Using a combination of exploratory and analytical methods, this study of machine learning and computer vision investigates and assesses current and prospective machine learning applications in computer vision. Research indicates that supervised, unsupervised, and semi-supervised machine learning algorithms are used in computer vision[5]. The most often used methods are Random Forest, neural networks, boosting and bagging, and support vector machines. The most cutting-edge applications of machine learning in computer vision are in the areas of object recognition, object classification, and relevant information extraction from images, graphics, and video[6], [7].

When it comes to computer vision, human position estimation is one of the most actively researched areas. The goal of dynamic pose estimation is to estimate human poses across all data sources. The process starts by viewing and mapping the skeletal coordinates. The divergence between the coordinates and proportional distance characteristics are used to determine what is contained inside the coordinates. Estimating where various human body components, such as the head, shoulder, elbow, wrist, hip, knee, and ankle, are located in a given picture is called human posture estimate. Sports, motion identification, character animation, clinical study of gait abnormalities, content-based video and picture

[1] *Department of CSE(AIML and CSBS), IEM Kolkata,*
deindrajit2016@gmail.com

[2]*Chitkara University Institute of Engineering and Technology,*
Chitkara University,, Punjab, India
lekhambhu@chitkara.edu.in

[3]*School of Computer Science and Engineering,*
RV University, Bengaluru, India.
rajatbhardwaj.x@gmail.com

[4]*Department of Computer Science and Engineering,*
School of Engineering and Technology, Sharda University,
Greater Noida, India
ambuj4u@gmail.com

[5]*Chitkara University Institute of Engineering and Technology,*
Chitkara University, Punjab, India
rajgaurang@chitkara.edu.in

* *Corresponding Author Email: ambuj4u@gmail.com*

retrieval, and intelligent video surveillance are just a few of the many essential applications of this basic topic in computer vision. Even after decades of study, estimating a stance is still challenging. Pose estimate is difficult because of the complexity of human articulation, which must be modeled.

1.1. Organization of paper

The remainder of the paper is structured as follows. In Section 2, we look at how machine learning has evolved and at some of the most useful applications of this technique to the categorization of body posture/activity. In Section 3, we presented the overarching notion of distribution-aware coordinate representation of body components that is helpful in posture categorization. In this part, we also provide some insight into the data sets, over-sampling techniques, and classifiers that were put to use. The experimental setup, including performance assessment measures and the statistical tests used here, is discussed in Section 4. Section 5 presents the findings of the experiments and discusses their significance. The last section, Section 6, summarises the key takeaways and suggests further paths of inquiry.

2. Literature Review

The detection of human body position has been the subject of several academic papers. Extracting information from databases is a common use of machine learning algorithms, and several have been developed, the most prominent of which are supervised learning methods. Recent research on this topic is surveyed in this section.

Using a combination of fuzzy logic and machine learning, Weiyan Ren et al.[8] demonstrated a data-efficient method for categorizing different sleeping positions of humans. A small dataset of 19,800 annotated depth photos captured by Kinect from 32 supine test individuals was used to train the classifier. Overall, 97.1 percent accuracy on the dataset was attained.

Xiaokang Zhou et al. [9] trained a classifier learning model to recognize human activities using weakly labeled sensor data. Intelligent auto labeling system based on deep Q-network (DQN) with distance-based incentive mechanism to increase IoT learning efficiency and manage improperly labeled samples. Long short-term memory (LSTM)-based classification is presented to recognize fine-grained patterns from sequential motion data, and a multisensor-based data fusion mechanism is devised to merge on-body sensor data, context sensor data, and personal profile data. The proposed technique was tested via experiments and assessments.

A wearable surface electromyography biosensing device with in-sensor adaptive learning capabilities has been disclosed by Ali Moin et al.[10]. For real-time gesture categorization, as well as for model training and updating under changeable circumstances like varied arm postures and sensor replacement, this system locally implements an algorithm for neuro-inspired hyperdimensional computing. When trained with a single trial per gesture, the system can correctly categorize 13 hand motions from two users with a 97.12 percent success rate. Maintaining a high level of accuracy (92.87%) even after increasing the number of gestures to 21 is possible thanks to the incorporation of model updates in response to changing conditions, which recovers an additional 9.5% of lost accuracy without requiring any extra computation on an external device.

Kenshi Saho et al. [11] developed a non-invasive, radar-based remote monitoring system for public restroom behavior and falls. The technique uses ceiling and wall-mounted Doppler radars. Doppler radar data is input into machine learning methods including CNNs, LSTMs, SVMs, and RFs to test the model's correctness and usefulness. A 21-person trial correctly categorized eight natural events, including falling. CNN worked best with Doppler spectrograms (time-velocity distribution), with 95.6% overall classification accuracy and 100% fall classification accuracy.

3. Material and Methods

3.1. Dataset

The dataset [12] used in this research contains the location of 33 pose landmarks (Fig 1).

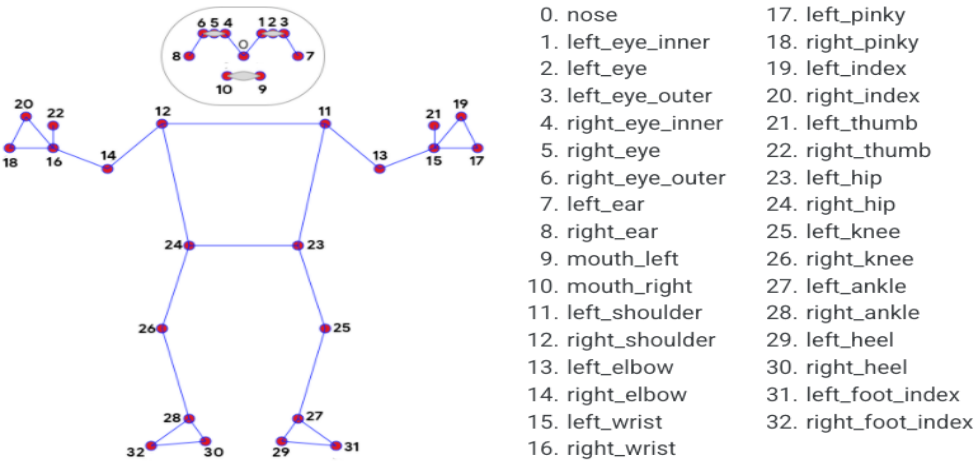

Fig. 1. Distribution-aware coordinate representation.

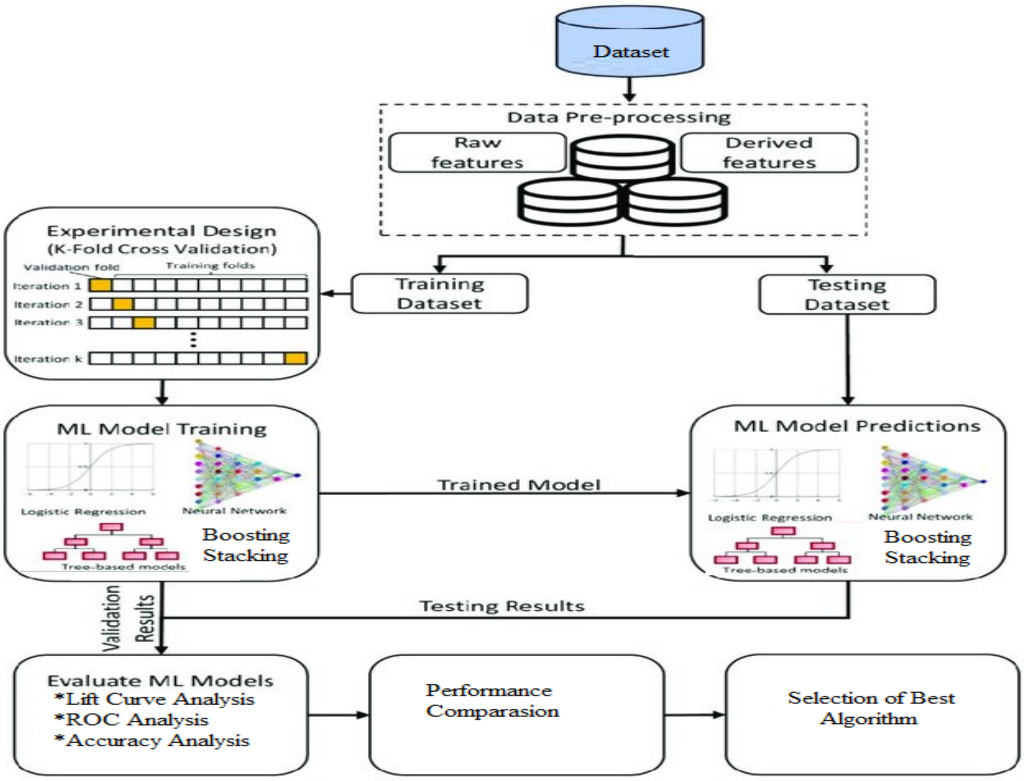

Fig. 2. Diagram showing how the Proposed Model would be put into action

Each landmark consists of the following:

- x and y: Points on the map have been scaled by the image's width and height to make them normalized to the range [0.0, 1.0].
- z: Specifies how far away from the camera a landmark is, using hip-center depth as the starting point; a lower number indicates a closer landmark. Z's magnitude is about comparable to x's.

3.2. Proposed Methodology

Fig. 2. depicts the progression of the implementation process.

3.2.1. Balancedness check for Dataset

In a multi-classification situation, it is common practice to ensure that each class is equally represented. Prior probability ratios across classes are highly skewed, as seen in Fig. 3. The issue of class divisions is the term for this predicament. When it comes to learning from unbalanced data, the "de facto" typical preparation approach is the Synthetic Minority Oversampling Technique (SMOTE)[13], [14]. This is because the technique is both easy to build and adapt to various types of situations. If you want to artificially swell the number of a minority group, the SMOTE algorithm is likely the method you'll see most often. Fig. 4 displays the dataset's even distribution after SMOTE resampling.

3.2.2. Dimensionality Reduction

A large number of dimensions are contained in the dataset, each one representing a particular axis along which the information may be read. Therefore, the more dimensions a dataset has, the more complicated it is to process the data. Complexity may be seen in terms of discovering and exploiting the links among distinct aspects of the dataset. Dimensionality reduction methods assist simplify relationships between several properties. These methods may be used to create a dataset with fewer dimensions than the one used as input. Dimensionality reduction is accomplished with the help of Principal Component Analysis in this study[15], [16]. Fig. 5 shows a scree plot, which is a depiction of the cumulative fraction of the variation explained. It's easy to see that if all 23 components are correct, we have 99.6 percent of the Explained variance.

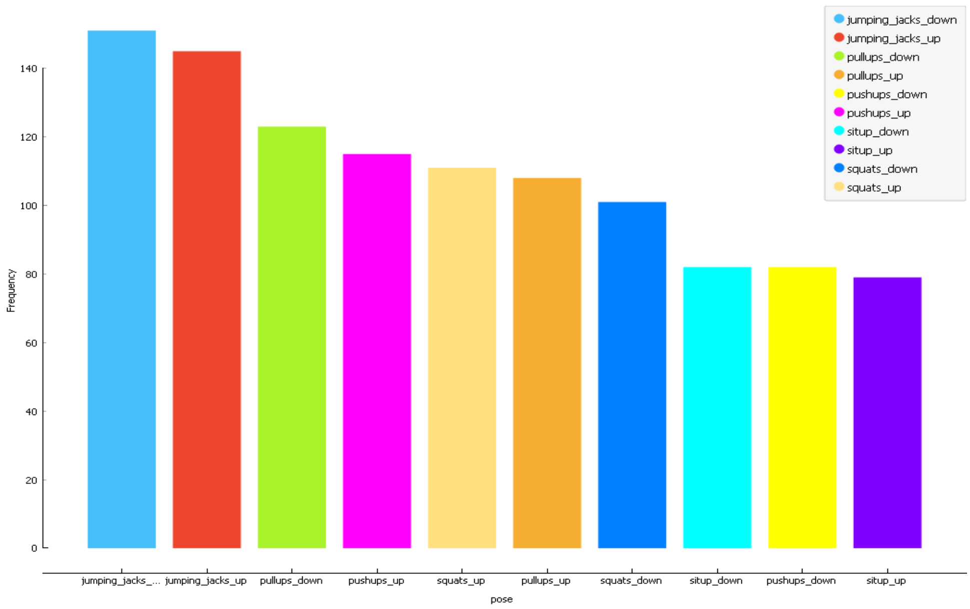

Fig. 3. The Distribution of the Intended Groups as a Count

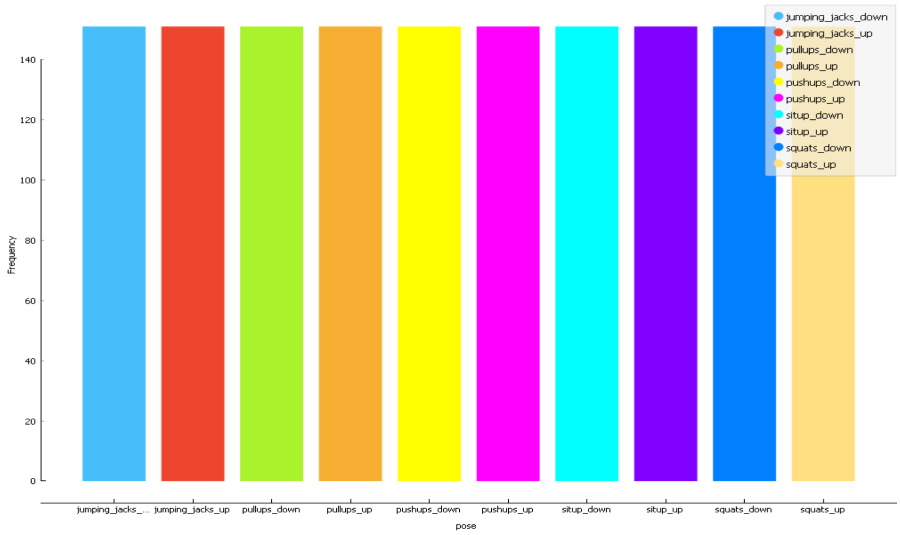

Fig. 4. SMOTE Resampling's Target Class Count Plot

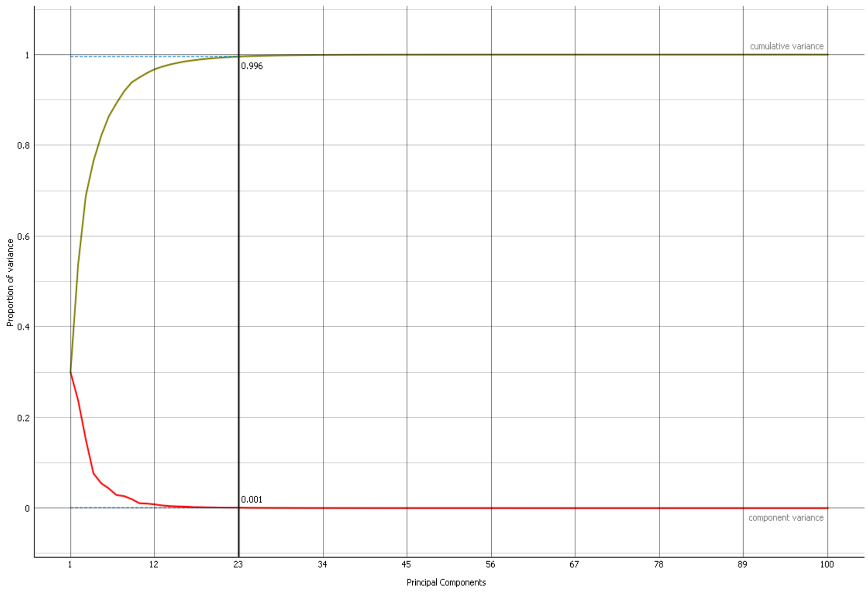

Fig. 5. Scree plot

3.2.3. Dataset Splitting

To increase the trustworthiness of the training sets and lessen the likelihood of any accidental traits being given more weight than they deserved, cross-validation methods were used for the datasets utilized in this work, which were of moderate size. Moreover, the prediction ability of an overfitted model is severely diminished on certain "unseen" testing data, which is a major drawback. Through a process of 10-fold cross-validation(CV), the dataset is split into training and testing sets. In 10-Fold CV, datasets are split into ten parts (or "folds"), with the first nine serving as test and training data, respectively. Information splitting is performed k times (k = 10) to ensure accuracy. The calculated performance averages are less likely to fluctuate when using 10-fold cross-validation (CV). The decision boundaries may be improved with the use of oversampling using the SMOTE technique, which has nothing to do with cross-validation. It does not just produce duplicate entries but rather interpolations depending on the minority group instead.

3.2.4. Training

We employ various widely used ML algorithms at this point, including basic machine learning approaches like Logistic Regression (LR), instance Based for K-Nearest Neighbor (KNN)[17], [18], Naive Bayes (NB)[19], [20], Random Forest (RF)[21], Decision Tree(DT)[22], Multi-Layer Perception (MLP) of Neural Network, and Support Vector Machine. We also applied boosting algorithms like Extreme Gradient Boost (XGboost), Gradient Boost[23], Adaboost[24], and Catboost. An ensemble approach like Stacking with two best classifiers was also applied. By combining the two models, we can achieve more accurate answers from most Stacked models. This is because various algorithms capture distinct patterns in the training data. Stacking's advantage is that it can pool the strengths of many high-quality models for a given classification to get predictions that outperform those of any one model in the ensemble.

3.2.5. Performance Evaluation

The area under the curve(AUC), Classification accuracy(ACC), Recall(REC), Precision(PRE), F1 score, and are only a few of the metrics that may be compared when assessing the efficacy of different machine learning algorithms via the use of confusion matrices. To further refine the evaluation of results, a ROC Curve Analysis is also carried out.

3.3. Experimental Setup

Core i7 processor, NVIDIA graphics card, 16 GB DDRAM, and 500 GB solid-state drive were used to conduct the research. The computer ran Microsoft's Windows 10. While developing with Python 3.9.7, the Anaconda IDE was used.

4. Results

It is important to highlight the machine learning methodologies used to get more accurate results while evaluating the efficacy of Machine Learning (ML) methods. We use metrics like accuracy, AUC, precision, F1-score, and recall to assess the efficacy of our model.

Fig. 6-19 show the confusion matrices showing the number of instances that were calculated using all of the models considered in this study.

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	121	2	1	7	0	1	0	1	3	15	151
jumping_jacks_up	7	104	26	9	0	0	0	1	0	4	151
pullups_down	1	39	92	2	2	2	5	5	1	2	151
pullups_up	7	5	17	100	0	2	0	3	6	11	151
pushups_down	1	0	2	0	134	12	1	0	1	0	151
pushups_up	0	1	0	3	11	133	3	0	0	0	151
situp_down	0	0	0	4	2	0	143	2	0	0	151
situp_up	2	5	4	6	0	0	3	118	12	1	151
squats_down	3	5	1	9	0	0	0	6	115	12	151
squats_up	26	13	4	23	1	2	0	2	2	78	151
∑	168	174	147	163	150	152	155	138	140	123	1510

Fig. 6. Confusion matrix for kNN

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	124	0	1	8	0	0	0	0	5	13	151
jumping_jacks_up	4	100	23	3	0	0	0	2	3	16	151
pullups_down	1	17	110	2	5	2	4	6	1	3	151
pullups_up	6	6	5	108	1	0	4	8	4	9	151
pushups_down	0	0	2	0	134	12	3	0	0	0	151
pushups_up	0	1	2	2	13	129	0	1	0	3	151
situp_down	0	0	2	4	2	3	134	5	1	0	151
situp_up	0	1	8	1	0	1	4	125	9	2	151
squats_down	2	1	6	3	1	0	1	9	123	5	151
squats_up	21	10	6	12	2	1	1	1	1	96	151
∑	158	136	165	143	158	148	151	157	147	147	1510

Fig. 7. Confusion Metrix of Decision Tree

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	126	1	1	4	0	0	0	0	1	18	151
jumping_jacks_up	1	121	13	4	0	0	0	1	3	8	151
pullups_down	0	11	124	3	4	2	4	3	0	0	151
pullups_up	2	2	2	119	1	0	3	9	4	9	151
pushups_down	0	0	0	0	140	7	3	0	0	1	151
pushups_up	0	0	1	0	6	139	1	0	2	2	151
situp_down	0	1	1	1	3	1	142	2	0	0	151
situp_up	1	2	6	0	1	0	1	133	6	1	151
squats_down	1	3	0	4	1	1	1	3	135	2	151
squats_up	11	5	2	7	0	2	2	1	0	121	151
∑	142	146	150	142	156	152	157	152	151	162	1510

Fig. 8. Confusion Metrix of RF

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	115	1	1	7	0	0	0	1	2	24	151
jumping_jacks_up	2	117	17	3	0	0	1	0	4	7	151
pullups_down	0	32	97	1	6	0	7	6	2	0	151
pullups_up	7	8	5	90	1	1	2	12	6	19	151
pushups_down	0	0	0	0	126	19	6	0	0	0	151
pushups_up	0	0	0	3	27	119	2	0	0	0	151
situp_down	0	0	0	2	7	0	140	2	0	0	151
situp_up	0	5	8	2	1	1	9	112	13	0	151
squats_down	0	1	1	15	0	0	7	8	119	0	151
squats_up	23	10	5	21	0	1	3	2	2	84	151
∑	147	174	134	144	168	141	177	143	148	134	1510

Fig. 9. Confusion Metrix of Naïve Bays

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	110	1	0	2	0	0	0	0	2	36	151
jumping_jacks_up	4	121	11	4	0	0	0	1	1	9	151
pullups_down	0	37	105	0	0	2	4	1	1	1	151
pullups_up	0	6	3	110	0	1	0	6	3	22	151
pushups_down	0	0	0	0	146	3	1	0	0	1	151
pushups_up	0	0	2	0	2	142	2	0	0	3	151
situp_down	0	0	1	0	3	0	145	2	0	0	151
situp_up	0	2	4	1	0	0	2	133	9	0	151
squats_down	1	0	1	6	0	0	0	2	141	0	151
squats_up	23	9	2	18	0	2	0	0	0	97	151
∑	138	176	129	141	151	150	154	145	157	169	1510

Fig. 10. Confusion Metrix of SVM

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	133	2	0	2	0	0	0	0	1	13	151
jumping_jacks_up	0	127	12	4	0	0	0	1	0	7	151
pullups_down	0	12	123	4	1	2	4	3	0	2	151
pullups_up	1	1	4	132	0	2	2	4	2	3	151
pushups_down	0	0	0	1	142	7	1	0	0	0	151
pushups_up	0	0	1	0	3	144	1	0	0	2	151
situp_down	0	1	0	2	2	0	143	3	0	0	151
situp_up	0	1	3	0	1	0	2	138	5	1	151
squats_down	0	0	1	4	0	0	0	4	142	0	151
squats_up	6	5	3	6	0	1	0	2	0	128	151
∑	140	149	147	155	149	156	153	155	150	156	1510

Fig. 11. Confusion Metrix of Catboost

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	115	1	0	10	0	0	0	0	2	23	151
jumping_jacks_up	2	112	15	4	0	1	0	1	2	14	151
pullups_down	0	12	114	7	6	2	3	5	0	2	151
pullups_up	7	3	1	110	0	2	1	5	8	14	151
pushups_down	0	3	1	2	129	13	2	0	1	0	151
pushups_up	1	1	1	4	9	131	2	0	1	1	151
situp_down	0	2	1	1	2	2	131	7	3	2	151
situp_up	0	5	5	2	2	1	1	128	5	2	151
squats_down	5	2	0	7	0	4	0	6	126	1	151
squats_up	14	9	3	14	0	2	1	1	3	104	151
∑	144	150	141	161	148	158	141	153	151	163	1510

Fig. 12. Confusion Metrix of XGBoostRF

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	119	2	0	6	0	0	0	0	3	21	151
jumping_jacks_up	2	110	20	3	0	0	1	3	0	12	151
pullups_down	0	16	108	7	2	1	4	5	0	8	151
pullups_up	11	3	5	104	1	0	2	6	9	10	151
pushups_down	0	0	1	2	136	7	4	1	0	0	151
pushups_up	2	2	2	1	9	131	0	0	1	3	151
situp_down	0	0	1	4	3	2	134	6	0	1	151
situp_up	0	2	6	4	0	0	4	125	8	2	151
squats_down	7	0	4	3	1	0	1	9	125	1	151
squats_up	16	7	5	8	1	1	0	2	0	111	151
∑	157	142	152	142	153	142	150	157	146	169	1510

Fig. 13. Confusion Metrix of AdaBoost

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	122	3	0	10	0	1	0	0	2	13	151
jumping_jacks_up	1	126	17	3	0	1	0	0	2	1	151
pullups_down	0	14	118	2	3	2	6	3	2	1	151
pullups_up	4	3	4	116	1	0	2	5	5	11	151
pushups_down	0	0	1	0	143	3	0	3	1	0	151
pushups_up	0	0	2	2	5	139	1	0	0	2	151
situp_down	0	0	4	1	1	0	144	1	0	0	151
situp_up	1	3	11	0	1	0	2	120	12	1	151
squats_down	3	1	0	13	1	1	1	14	115	2	151
squats_up	25	16	2	24	0	2	0	0	0	82	151
∑	156	166	159	171	155	149	156	146	139	113	1510

Fig. 14. Confusion Metrix of SGD

Actual \ Predicted	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	128	1	0	3	0	2	0	0	1	16	151
jumping_jacks_up	2	119	13	4	1	0	0	1	1	10	151
pullups_down	2	15	116	2	2	1	3	8	0	2	151
pullups_up	3	3	2	116	0	0	3	6	11	7	151
pushups_down	0	1	0	0	146	2	1	1	0	0	151
pushups_up	2	0	0	1	2	142	1	0	0	3	151
situp_down	0	1	3	3	1	1	138	1	3	0	151
situp_up	0	7	3	0	0	0	2	131	8	0	151
squats_down	2	0	1	4	1	1	1	9	131	1	151
squats_up	16	8	1	8	1	1	0	0	0	116	151
∑	155	155	139	141	154	150	149	157	155	155	1510

Fig. 15. Confusion Metrix of LR

Predicted

Actual	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	130	3	0	2	0	0	0	0	1	15	151
jumping_jacks_up	0	122	16	4	0	0	0	0	0	9	151
pullups_down	0	12	126	3	1	1	3	2	1	2	151
pullups_up	3	2	2	128	1	0	1	5	2	7	151
pushups_down	0	0	0	0	146	1	2	2	0	0	151
pushups_up	0	0	1	0	3	143	1	0	0	3	151
situp_down	0	0	2	2	2	0	142	2	1	0	151
situp_up	0	1	5	0	1	0	2	137	4	1	151
squats_down	2	0	0	3	1	0	0	2	141	2	151
squats_up	10	7	1	13	0	2	0	0	0	118	151
∑	145	147	153	155	155	147	151	150	150	157	1510

Fig. 16. Confusion Metrix of GB

Predicted

Actual	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	132	0	0	2	0	0	0	0	1	16	151
jumping_jacks_up	0	126	11	1	0	0	1	0	1	11	151
pullups_down	0	7	132	3	0	1	2	4	1	1	151
pullups_up	2	3	1	133	1	1	1	4	1	4	151
pushups_down	0	0	1	0	147	2	1	0	0	0	151
pushups_up	1	0	1	0	1	145	1	0	0	2	151
situp_down	0	1	2	2	1	0	143	2	0	0	151
situp_up	0	0	1	1	0	0	3	142	4	0	151
squats_down	1	1	1	1	0	0	1	1	142	3	151
squats_up	10	11	0	4	0	2	1	0	0	123	151
∑	146	149	150	147	150	151	154	153	150	160	1510

Fig. 17. Confusion Metrix of MLP(NN)

Predicted

Actual	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	132	0	0	2	0	0	0	0	1	16	151
jumping_jacks_up	0	130	8	1	0	0	0	1	1	10	151
pullups_down	0	7	133	4	0	1	2	3	0	1	151
pullups_up	1	0	2	138	0	2	1	3	0	4	151
pushups_down	0	0	1	0	146	3	1	0	0	0	151
pushups_up	1	0	1	0	1	147	0	0	0	1	151
situp_down	0	1	2	2	1	0	143	2	0	0	151
situp_up	0	1	1	1	1	0	1	143	3	0	151
squats_down	1	2	1	2	0	0	0	1	144	0	151
squats_up	5	6	0	5	0	2	1	0	0	132	151
∑	140	147	149	155	149	155	149	153	149	164	1510

Fig. 18. Confusion Metrix of Stack

Predicted

Actual	jumping_jacks_down	jumping_jacks_up	pullups_down	pullups_up	pushups_down	pushups_up	situp_down	situp_up	squats_down	squats_up	∑
jumping_jacks_down	131	1	0	1	0	1	0	0	1	16	151
jumping_jacks_up	1	125	14	3	0	0	0	0	2	6	151
pullups_down	0	7	130	2	2	0	3	4	1	2	151
pullups_up	3	2	3	125	0	1	3	7	2	5	151
pushups_down	0	0	0	0	145	2	1	2	0	1	151
pushups_up	0	0	1	0	4	144	0	0	0	2	151
situp_down	0	0	2	2	1	0	143	2	1	0	151
situp_up	0	1	4	0	1	0	2	138	4	1	151
squats_down	0	0	1	3	1	0	0	3	143	0	151
squats_up	6	6	3	8	0	2	0	0	0	126	151
∑	141	142	158	144	154	150	152	156	154	159	1510

Fig 19. Confusion Metrix of XGBoost

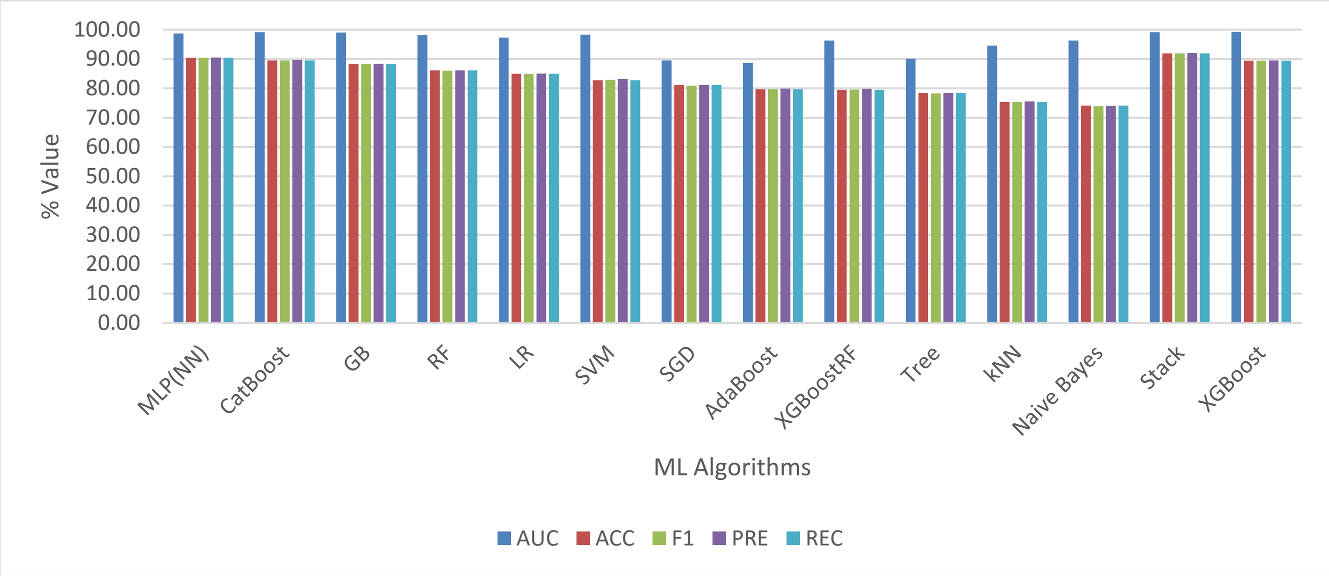

Fig. 20. Classification Performance Comparision

As can be seen in Fig. 18, proper classification instances are most common and misclassification cases are least common when compared across all methods. Tables I provide the ACC, PRE, REC, F1 score, and area under the curve (AUC) of different classification methods, which correspond to the confusion metrics shown in Fig. 6–19.

TABLE I. COMPARATIVE ANALYSIS OF PERFORMANCE

ML Algorithms	AUC	ACC	F1	PRE	REC
MLP(NN)	98.75	90.40	90.42	90.47	90.40
CatBoost	99.12	89.54	89.54	89.59	89.54
GB	99.02	88.28	88.31	88.36	88.28
RF	98.11	86.09	86.08	86.13	86.09
LR	97.33	84.97	84.95	85.00	84.97
SVM	98.27	82.78	82.86	83.23	82.78
SGD	89.51	81.13	80.93	81.15	81.13
AdaBoost	88.70	79.67	79.73	79.89	79.67
XGBoostRF	96.30	79.47	79.57	79.76	79.47
Tree	90.02	78.34	78.29	78.35	78.34
kNN	94.59	75.36	75.32	75.59	75.36
Naive Bayes	96.30	74.11	73.89	74.03	74.11
Stack	99.18	91.92	91.95	92.04	91.92
XGBoost	99.24	89.40	89.41	89.48	89.40

A comparison of the degrees to which various machine learning approaches achieve accurate categorization is shown in Fig. 20.

As can be seen in Fig. 20, the Stacking with Catboost and MLP algorithm achieves an impressive 91.92% accuracy. The Precision, Recall, and F1-score all corroborate this.

5. Analysis

5.1. ROC Curve Analysis

The receiver operating characteristics (ROC) curve is one of the most important evaluation metrics for probabilities. It illustrates the relationship between the true positive rate (TPR) and the false positive rate (FPR) as the decision threshold shifts, which enables an accurate assessment of which classification model is the most effective. The ROC, in conjunction with the AUC, which is a measure of separability, illustrates the degree to which an algorithm discriminates between different groups. A well-performing model will have a high AUC. According to the ROC curve, a perfect classification is represented by the point on the

TPR vs. FPR graph that has coordinates of 0 and 1 in the top left-hand corner of the graph (Fig. 21).

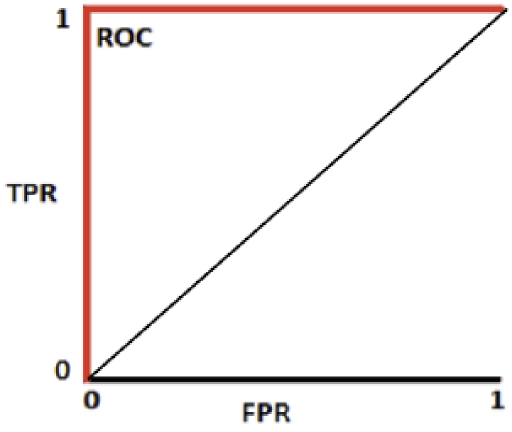

Fig 21. Receiver operating Curve

Fig. 22-24 show ROC Curves for three different target classes respectively.

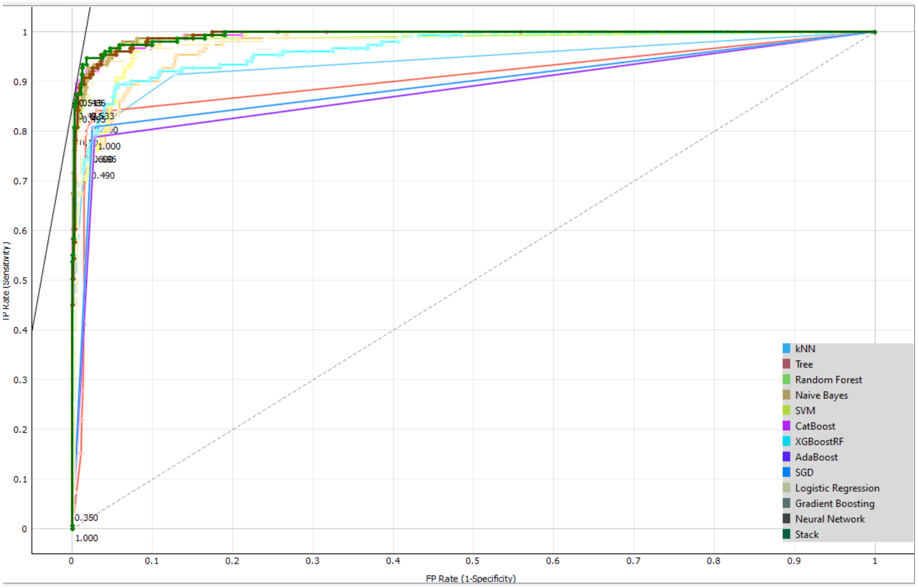

Fig. 22. ROC class analysis: Jumping_Jacks_down

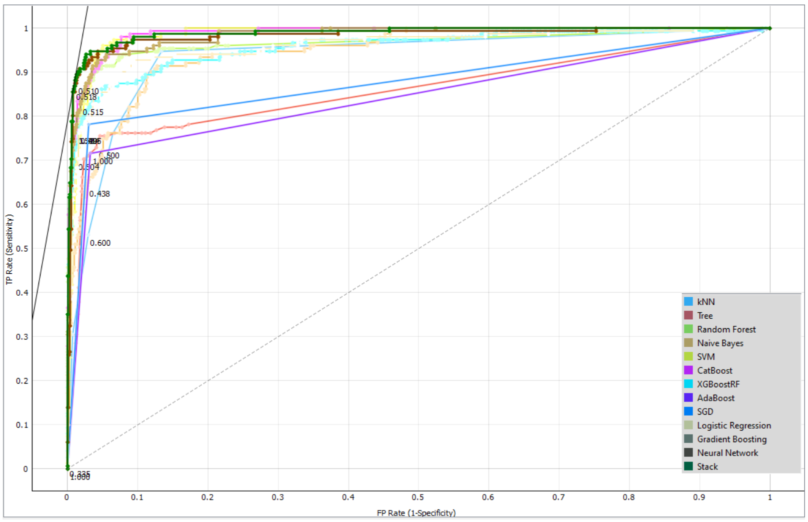

Fig. 23. ROC class analysis: Pull_up_down

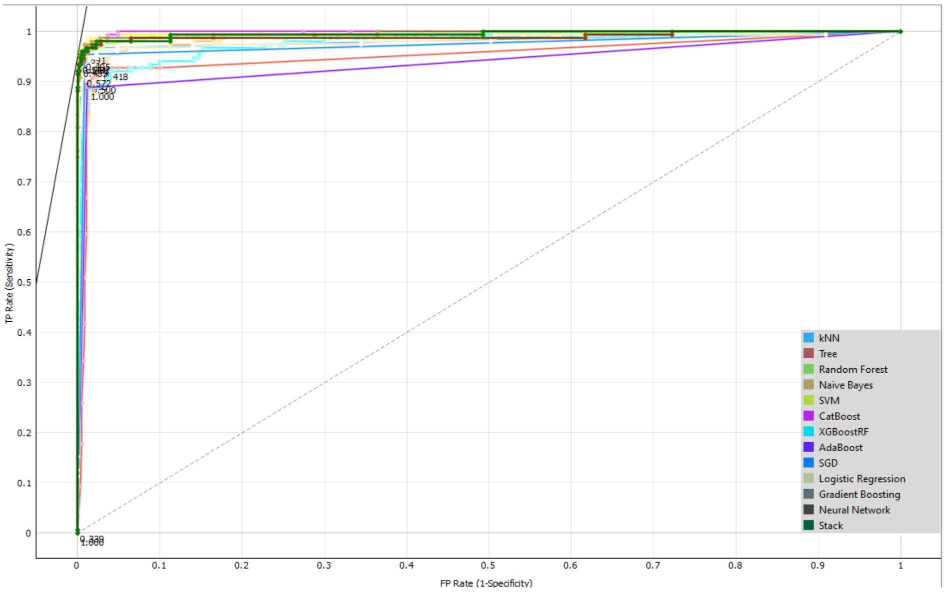

Fig. 24. ROC class analysis: Sit_up_down

The overall practical implications of the cross-validation performances and the testing findings are shown in Fig 22-24. When compared to other common machine learning methods, the Stack approach offers the greatest average ROC.

6. Conclusion

Computer vision and machine learning aim to include human-level data perception, interpretation, and action-taking. Research in machine learning and computer vision is ongoing. Computer vision is crucial for IoT, IIoT, and brain-computer interfaces. Using machine learning and computer vision, we can recognize and monitor human activities in real time across media sources. Traditional methods of generating informed estimates and completing extensive analysis include supervised, unsupervised, and semi-supervised learning. These strategies leverage machine learning algorithms like SVM and KNN. Collecting data, training a model, and implementing its predictions are machine learning solutions.

This study offered a viable paradigm for diagnosing and predicting breast cancer. Our proposed system for breast cancer detection includes data collection, pre-processing, segmentation, hyper-parameter tuning for ML models, classification based on ML, and prediction and assessment metrics. We analyzed seven machine learning techniques and boosting algorithms, including XGboost, Multi-Layer Perception, Naive Bayes, KNN, and Random Forest. The performance of frameworks' classifiers on the posture dataset was compared. Classifications were also stacked. Two-level estimators form the structure. First layer contains all baseline models used to predict test dataset outcomes. A Meta-Classifier synthesizes new predictions from baseline model output. We used catboost and Multi-Layer Perception for stacking which gave the best accuracy (91.92%).

To evaluate the efficacy of the algorithm in a real-world scenario, the authors plan to undertake a series of onsite experiments with a bigger sample size in the near future. It is also our intention to use swarm algorithms to determine optimal values for the hyperparameters. Additionally, we will adapt the selected models into a workable and usable resource for helping doctors make accurate diagnoses of breast cancer. Additional machine learning methods, new forms of physical exercise, and different kinds of datasets might all be investigated in further studies.

Compliance with Ethical Standards:

Funding: No Funding

Conflict of Interest: The authors declare that they have no conflicts of interest to report regarding the present research.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

[1] N. K. Trivedi, V. Gautam, H. Sharma, A. Anand, and S. Agarwal, "Diabetes Prediction using Different Machine Learning Techniques," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022, pp. 2173–2177, 2022, doi: 10.1109/ICACITE53722.2022.9823640.

[2] N. Ujjwal, A. Singh, A. K. Jain, and R. G. Tiwari, "Exploiting Machine Learning for Lumpy Skin Disease Occurrence Detection," 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 1–6, Oct. 2022, doi: 10.1109/ICRITO56286.2022.9964656.

[3] I. Al-muwaffaq and Z. Bozkus, "MLTDD: use of machine learning techniques for diagnosis of thyroid gland disorder," Comput Sci Inf Technol, pp. 67–73, 2016, doi: 10.5121/csit.2016.60507.

[4] A. K. Agarwal, V. Kiran, R. K. Jindal, D. Chaudhary, and R. G. Tiwari, "Optimized Transfer Learning for Dog Breed Classification," International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 1s, pp. 18–22, Oct. 2022, Accessed: Jan. 08, 2023. [Online]. Available: https://www.ijisae.org/index.php/IJISAE/article/view/2233

[5] R. G. Tiwari, D. S. Yadav, and A. Misra, "Performance Evaluation of Optimizers in the Classification of Marble Surface Quality Using CNN," pp. 181–191, 2023, doi: 10.1007/978-981-19-3148-2_15/COVER.

[6] R. G. Tiwari, A. K. Agarwal, R. K. Kaushal, and N. Kumar, "Prophetic Analysis of Bitcoin price using Machine Learning Approaches," in Proceedings of IEEE International Conference on Signal Processing,Computing and Control, 2021, vol. 2021-Octob. doi: 10.1109/ISPCC53510.2021.9609419.

[7] M. M. Panda, S. N. Panda, and P. K. Pattnaik, "Exchange Rate Prediction using ANN and Deep Learning Methodologies: A Systematic Review," Indo - Taiwan 2nd International Conference on Computing, Analytics and Networks, Indo-Taiwan ICAN 2020 - Proceedings, pp. 86–90, Feb. 2020, doi: 10.1109/INDO-TAIWANICAN48429.2020.9181351.

[8] W. Ren, O. Ma, H. Ji, and X. Liu, "Human Posture Recognition Using a Hybrid of Fuzzy Logic and Machine Learning Approaches," IEEE Access, vol. 8, pp. 135628–135639, 2020, doi: 10.1109/ACCESS.2020.3011697.

[9] X. Zhou, W. Liang, K. I. K. Wang, H. Wang, L. T. Yang, and Q. Jin, "Deep-Learning-Enhanced Human Activity Recognition for Internet of Healthcare Things," IEEE Internet Things J, vol. 7, no. 7, pp. 6429–6438, Jul. 2020, doi: 10.1109/JIOT.2020.2985082.

[10] A. Moin et al., "A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition," Nature Electronics 2020 4:1, vol. 4, no. 1, pp. 54–63, Dec. 2020, doi: 10.1038/s41928-020-00510-8.

[11] K. Saho, S. Hayashi, M. Tsuyama, L. Meng, and M. Masugi, "Machine Learning-Based Classification of Human Behaviors and Falls in Restroom via Dual Doppler Radar Measurements," Sensors 2022, Vol. 22, Page 1721, vol. 22, no. 5, p. 1721, Feb. 2022, doi: 10.3390/S22051721.

[12] "Physical Exercise Recognition Dataset | Kaggle." https://www.kaggle.com/datasets/muhannadtuameh/exercise-recognition (accessed Jan. 08, 2023).

[13] V. Rattan, R. Mittal, J. Singh, and V. Malik, "Analyzing the application of SMOTE on machine learning classifiers," 2021 International Conference on Emerging Smart Computing and Informatics, ESCI 2021, pp. 692–695, Mar. 2021, doi: 10.1109/ESCI50559.2021.9396962.

[14] A. Fernández, S. García, F. Herrera, and N. v. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," Journal of Artificial Intelligence Research, vol. 61, pp. 863–905, Apr. 2018, doi: 10.1613/JAIR.1.11192.

[15] F. Kherif and A. Latypova, "Principal component analysis," Machine Learning: Methods and Applications to Brain Disorders, pp. 209–225, Jan. 2020, doi: 10.1016/B978-0-12-815739-8.00012-2.

[16] R. Vidal, Y. Ma, and S. S. Sastry, "Principal Component Analysis," pp. 25–62, 2016, doi: 10.1007/978-0-387-87811-9_2.

[17] W. Xing and Y. Bei, "Medical Health Big Data Classification Based on KNN Classification Algorithm," IEEE Access, vol. 8, pp. 28808–28819, 2020, doi: 10.1109/ACCESS.2019.2955754.

[18] M. Goswami and N. J. Sebastian, "Performance Analysis of Logistic Regression, KNN, SVM, Naïve Bayes Classifier for Healthcare Application During COVID-19," Lecture Notes on Data Engineering and Communications Technologies, vol. 96, pp. 645–658, 2022, doi: 10.1007/978-981-16-7167-8_47/COVER.

[19] S. Chen, G. I. Webb, L. Liu, and X. Ma, "A novel selective naïve Bayes algorithm," Knowl Based Syst, vol. 192, p. 105361, Mar. 2020, doi: 10.1016/J.KNOSYS.2019.105361.

[20] F. J. Yang, "An implementation of naive bayes classifier," Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018, pp. 301–306, Dec. 2018, doi: 10.1109/CSCI46756.2018.00065.

[21] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," Expert Syst Appl, vol. 134, pp. 93–101, Nov. 2019, doi: 10.1016/J.ESWA.2019.05.028.

[22] Priyanka and D. Kumar, "Decision tree classifier: A detailed survey," International Journal of Information and Decision Sciences, vol. 12, no. 3, pp. 246–269, 2020, doi: 10.1504/IJIDS.2020.108141.

[23] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," Artificial Intelligence Review 2020 54:3, vol. 54, no. 3, pp. 1937–1967, Aug. 2020, doi: 10.1007/S10462-020-09896-5.

[24] P. Bahad and P. Saxena, "Study of AdaBoost and Gradient Boosting Algorithms for Predictive Analytics," pp. 235–244, 2020, doi: 10.1007/978-981-15-0633-8_22.

[25] Ambuj Kumar Agarwal, Gulista Khan, Shamimul Qamar, Niranjan Lal,Localization and correction of location information for nodes in UWSN-LCLI,Advances in Engineering Software,Volume 173,2022,103265,ISSN 0965-9978,https://doi.org/10.1016/j.advengsoft.2022.103265

[26] R. Sharma, H. Pandey, and A. K. Agarwal, "Exploiting artificial intelligence for combating COVID-19 : a review and appraisal," vol. 12, no. 1, pp. 514–520, 2023, doi: 10.11591/eei.v12i1.4366.

[27] D. Srivastava, H. Pandey, and A. K. Agarwal, "Complex predictive analysis for health care : a comprehensive review," vol. 12, no. 1, pp. 521–531, 2023, doi: 10.11591/eei.v12i1.4373.

[28] A. K. . Agarwal, V. . Kiran, R. K. . Jindal, D. . Chaudhary, and R. G. . Tiwari, "Optimized Transfer Learning for Dog Breed Classification", Int J Intell Syst Appl Eng, vol. 10, no. 1s, pp. 18–22, Oct. 2022.