International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

AI Assisted Human Activity Recognition (HAR)

Sunita Kumari Chaurasia, S.R.N Reddy

Abstract: ***Human Activity Recognition and assisting user on the basis of his context is attracting researchers since decade Researchers are working in the area to increase the accuracy of detection by various means. The challenging issue is to determine the correct supervised classifier for the detection purpose. This paper intent to examine the methodology used to recognize HAR and the impact of classifiers practiced in training and Testing. We have also tried to identify the suitable supervised machine learning model for HAR. Data of 30 Users with 561 features belonging to accelerometer and gyroscope sensor of smartphone from UCI repository is used for evaluation purpose. Nine different supervised machine learning Models are trained and tested on the dataset. The result concludes that HAR is a process which depends upon the classifiers used. It also conclude that out of 9 different Machine learning models ANN performs well and after that SVM, kNN, Random Forest and Extra Tree are equally good models for the purpose of HAR with Accuracy and execution time as the performance evaluation metric.***

Keywords: ***Activity Recognition, Classification, Supervised learning, Machine Learning Models.***

I. INTRODUCTION

Sensing User context with the help of Artificial Intelligence in order to facilitate and assist is trending research area. Human Activity recognition (HAR) is one of the dimensions which help in detecting context of the user. . It is an area which binds sensor technology and Artificial Intelligence together. HAR plays essential role in our daily life monitoring. The various application area of HAR is healthcare, transportation, life logging, fitness tracking, monitoring physical activities of elderly people, security and surveillance etc.

HAR is mainly comprise of four steps, first gathering data from sensors, second extracting required features, third training the model and fourth predicting the unknown input. All the steps are of equal priority, however opting correct Artificial Intelligence subset i.e classification model for training and prediction is of great importance. . Finding the context accurately and providing the user or application in lesser time is the research focus. However the impact of different classifiers with respect to HAR has not been explored much. We identified this gap and tried to provide user all details at one place. This paper provides candidature of machine learning (ML) models for HAR on the basis of the performance of various models on benchmark dataset keeping accuracy and execution time as evaluation metric.

Rest of the paper is organized as follows: section 2 provides brief introduction of HAR process, section 3 is focusing on various classification techniques used in HAR and Section 4 shows experiments and evaluation done on the UCI dataset using various ML algorithms and section 5 concerns conclusion and future scope.cation is not possible.

II. HAR PROCESS

Author Human Activity recognition is a process comprises of four main steps shown in figure 1. The first Step is Data collection phase where the inertial mobile or wearable sensors are used for collecting raw data. The sensors could be of various types like Accelerometer, Gyroscope, Magnetometer, Proximity sensors belonging to Motion category, Heart rate, Respiration Rate, SpO2, Finger Temperature etc. belonging to Physiological sensors taking data of user's body, GPS belonging to sensing user location and Pressure, Humidity, Temperature, smoke sensors belonging to environmental sensors category. The second step involved in HAR process is feature extraction. The raw data is processed to generate various time domain and frequency domain features. The main aim of feature extraction is dimensionality reduction of the original set of raw data, whose derivatives are expected to add new information in knowledge. The third step is Model training, in which different classification models are applied on the labeled dataset, this is called supervised learning. In this the model is trained with the help of examples. On the other side when the categorization is done without any help of labeled data, it is called unsupervised learning. K Mean clustering, Hidden Markov Model (HMM), Gaussian Mixture Models (GMMs) belongs to unsupervised learning however Decision Tree (DT), Support Vector Machine (SVM), Naïve Bays (NB), k Nearest Neighbour (KNN), Artificial Neural Networks (ANN), Random Forest Classifier (RFC), etc. belongs to supervised learning.

III. CLASSIFICATION TECHNIQUES

The classification techniques are also known as supervised ML models. Mostly used supervised classifiers in HAR are Decision Tree, Random Forest, Support Vector Machine, K nearest neighbour, Naïve Bays, Multi-Layer Perceptron etc. The Classifiers which have been used in the experiments for training and prediction purpose are concisely described below:

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Sunita Kumari Chaurasia, CSE Dept, IGDTUW, Delhi, India.
S.R.N Reddy, CSE Dept, IGDTUW, Delhi, India.

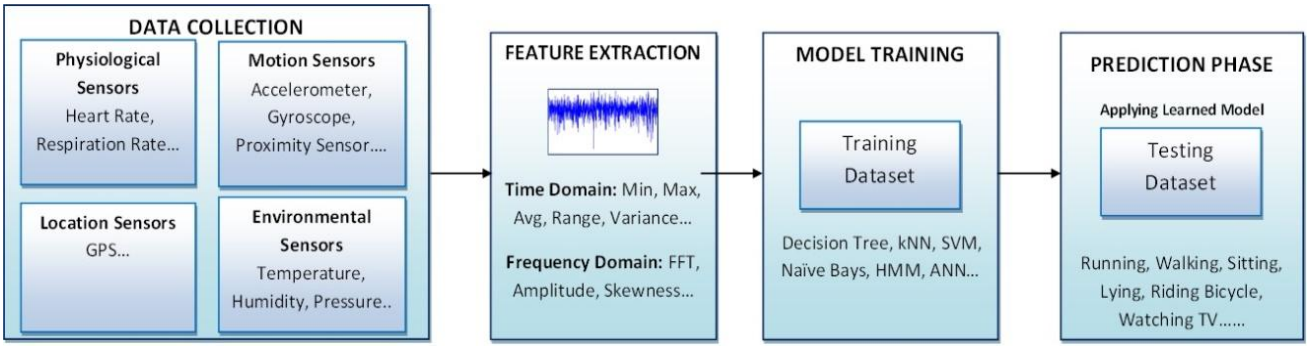

Fig. 1 HAR Process

A. Decision Tree Classifier (DT)

It is the simplest hierarchical flow chart type algorithm, which uses branching method to illustrate every possible outcome of a decision [1]. Internal nodes are for testing an attribute, the outcome of the test is shown by branches and leaves symbolize the class label [2]. The decision process starts from any selected attribute node n which has a test functionf_n (x); test function is applied to the input and according to the result one of the corresponding branch is selected. This process gets continued till it reaches to the leaf node, this leaf node indicates the class to which that input data belongs [3]. The aim of this process is to find the smallest possible tree. The promising attribute detection for starting the split is done by a purity measure function known as Entropy or gini impurity, it is used to calculate the information gain. The information gain for selecting an attribute A is calculated by using equation 1 and the attribute with maximum gain is selected for splitting [4].

$$Gain(A) = Info(D) - Info_A(D) \qquad (1)$$

B. K Nearest Neighbors Classifier (KNN)

It is a supervised lazy learner technique which learns by analogy. The training instances are represented in multiple dimension space. When any unknown tuple arrives, k-nearest neighbor classifier searches the k training instances, which are in the proximity of that unknown tuple. Then on the basis of majority of votes it places this unknown tuple to the nearest class[5]. These k preparing instances are the k "closest neighbors" [6]. This technique is sensitive to the value of k. If the value of k increases the prediction time also increases. "Closeness" between two tuples X_1 and X_2 belonging to n dimensional space can be determined by using any distance metric like Euclidean distance [7] given by equation:

$$\text{dist}(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2} \qquad (2)$$

C. Support Vector Machine Classifier (SVM/SVC)

It is a supervised ML technique where training data is transformed into n dimension space, and then SVC searches for optimal linear hyperplane which separates the data into different classes. This hyperplane can be understood as a "decision boundary"[8]. SVC uses the essential training tuples (support vectors) and margins to find its hyperplane. There can be infinite number of hyperplane between two classes but the algorithm try to discover the hyperplane with maximum margin [9]. The training time of SVC algorithms are more, but they give accurate results. They are less susceptible to overfitting. The maximum margin hyperplane is determined by equation below where C_i represents class label of support vector X_i; testing input X^T; Lagrangian multipliers αi and b0 are numeric parameters which is automatically calculated by optimizer or SVM; n represents no of support vectors:

$$d(X)^T = \sum_{i=1}^{n} C_i \propto_i X_i X^T + b_0 \qquad (3)$$

D. Naïve Bays (NB)

It is a probabilistic based technique, which learns likelihood of an object with definite features belonging to particular class. A posterior probability of every class is calculated by estimating conditional probabilities from the training input. It can also be represented using a simple Bayesian network[10]. It is called naïve because it makes assumption that the existence of one feature is not related to the existence of another feature[11].Bayesian Rule is given as:

$$(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \qquad (4)$$

Here P (X|Y) is Posterior Probability of occurrence of event X when Y is valid. $P(Y|X)$ is conditional probability of Y given X, P(X) and P(Y) are Prior Probability of occurrence of event X and Y.

E. Random Forest Classifier (RFC)

RFC is ensemble of Decision tree classification. Multiple decision trees are built at the time of training and result of the class is on the basis the class which corresponds to highest number of occurrence i.e majority voting.

The input dataset is divided into multiple subsamples randomly and trees are formulated from each random samples [12]. These trees are non-correlated as they have split on different features (features are also random). For testing a new tuple these different trees are used and it passes the input tuple through these sub trees existing in the forest. Each tree contributes "votes" for some class or each tree classifies the input vector and put it into some class.

The forest with the highest vote is being chosen as its final class[11].

F. Ada Boost Classifier (ABC)

Adaptive Boosting is another name of AdaBoost, It focuses on converting a set of weak classifiers into a robust one [11]. The whole dataset is divided into multiple samples and different classifiers are applied on them. Here consequent weak learners are boosted with respect to misclassified instances classified by former classifier. Initially equal probability or weights are assigned to dataset, but as per result of classification higher weights are assigned to misclassified dataset so that they can appear in next training subset. The outputs of different classifiers are combined by majority voting, the weight is assigned to the classifiers based on their accuracy[13]. Classifier with 50% accuracy is given a weight of zero, classifier with less than 50% is given negative weights and greater than 50% are assigned positive weights. The algorithm hunts for appropriate weights distribution beginning from the uniform distribution of weights. To calculate the error rate of learnerL_i, weights of every tuples is summed from dataset $D_i$which was misclassified byL_i. That is,

$$\text{error}(L_i) = \sum_{i=1}^{d} w_j \times \text{err}(X_i) \quad (5)$$

Here $\text{err}(X_i)$is the misclassification error for instance(X_i); when it is misclassified, $\text{err}(X_i)$is 1; else, it is 0.

G. Bagging Classifier

Bagging is another name for Bootstrap aggregating [14]. It is an ensemble machine learning algorithm which improves the stability and accuracy of weak ML algorithms. Here many random sub-samples of main dataset are created with replacement (means same value can be selected multiple times). It is similar to Random Forest with a difference that the original training dataset is sub-sampled with replacement; some instance or tuples may be repeated many times while others are left out. Then similar or different ML algorithms are applied on different subsamples, and the output of each is averaged out. It is used to reduce variance and helps in avoiding overfitting. It is a method for the averaging of output of various models.

H. Extra Tree Classifier (ETC)

Extra-Tree classifier is also known as Extremely Randomized trees. The chief objective of this classifier is to further randomize the technique of building a tree in the perspective of input features [15]. The chief variances with former tree based ensemble techniques are that first it choose the cut-points completely random for splitting the nodes rather than finding best cut-point and second it usages the entire learning sample instead of bootstrap replica to produce the trees. It increases accuracy and reduces computational burdens linked with the determination of optimal cut-points as in Decision trees or random forests classifiers.

I. Artificial Neural Network (ANN) or MLP Classifier (MLP):

The multilayer perceptron is a feed forward or back propagation ANN model that binds sets of input data onto a set of suitable outputs [16]. It imitates the characteristics of human brain i.e neuron. An MLP consists of several layers and previous layer is completely connected to the subsequent layer [17]. Except the input node all nodes works as neurons using some nonlinear activation functions. There can be one or more non-linear hidden layers between the input and the output layer. The output at a hidden layer is calculated by:

$$y_i = f\left(\sum_{j=1}^{m} w_{ij} x_j + b_i\right) \quad (6)$$

Where w_{ij}weight from j is input node to i hidden node, x_j is input from
j node, and b_i is the bias at activation node.

IV. EXPERIMENT AND ANALYSIS

The methodology used for evaluation purpose is shown in fig. 2. It is comprised of various steps:

A. Sensor Data and Feature Extraction

We have used UCI HAR dataset for the analysis purpose [18]. The dataset provided contains data for 6 activities from 30 people. The dataset is provided in two files, one for training and another for testing purpose. The distribution of data is shown in table 1. However we have combined the train and test dataset and then randomly divided it into ratio of 70:30 to see the output of subject dependent data.

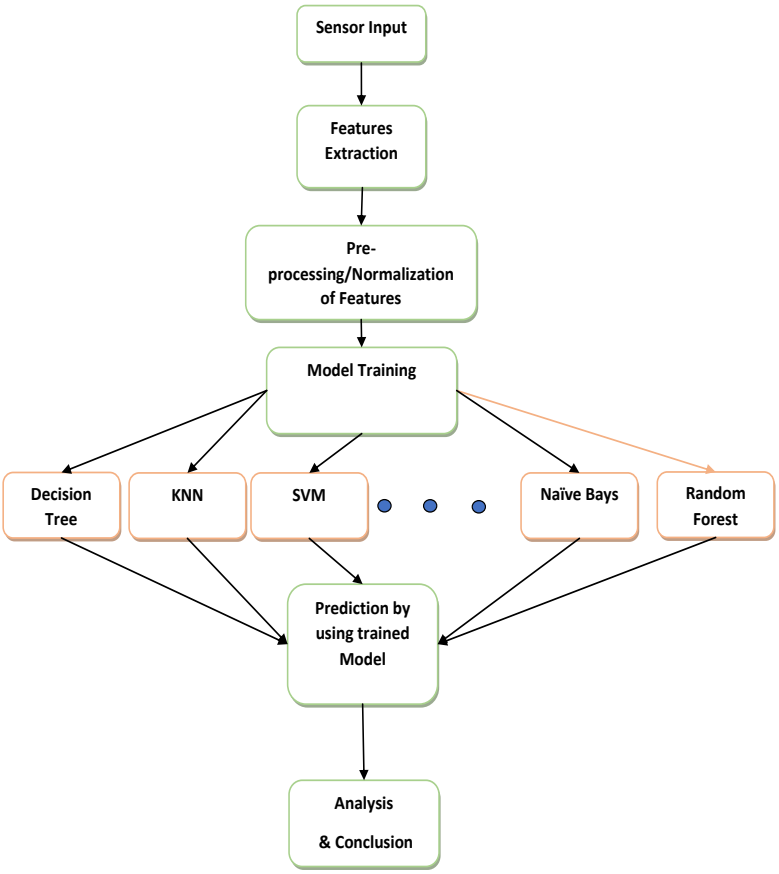

Fig. 2. Proposed Methodology

Table 1 UCI Dataset Information

Sr. No	No of Activity	No of users	Sensors used	No of attributes/features	No. of Train data	No. of Test Data	Total No. of data
1	6	30	Accelerometer & Gyroscope	561	7352	2947	10299

B. Pre-processing

These features were normalized by using pre-processing libraries of sklearn in python. It Standardize features by removing the mean and scaling to unit variance by using formulae:

$$z = \frac{x-\mu}{s} \quad (7)$$

Where x is the original value, μ is the mean and s is standard deviation. This is done so that a feature having larger order of variance than others should not dominate the objective function.

C. Model Training

We have used nine different classifiers for the training purpose (as mentioned in section III). The 70 % of normalized data are used to trined these 9 supervised classifiers. Jupiter Notebook with Scikit learn libraries are used for the training and testing the models.

D. Prediction using trained model

The above trained models are applied on 30% of remaining dataset for class prediction. The parameters used for performance evaluation are accuracy and execution time. The experiment was executed 10 times to reduce the bias and the mean accuracy and mean execution time was calculated. The results achieved by applying above models are shown in figure 3 and 4. In fig 3 X axis portrays the different models, whereas the Y axis represents the Accuracy score in a scale of 0 to 1 where 1 represents 100% accuracy. In fig 4 X axis depicts the different models, whereas the Y axis represents the execution time in sec. higher accuracy with low execution time will be the suitable model.

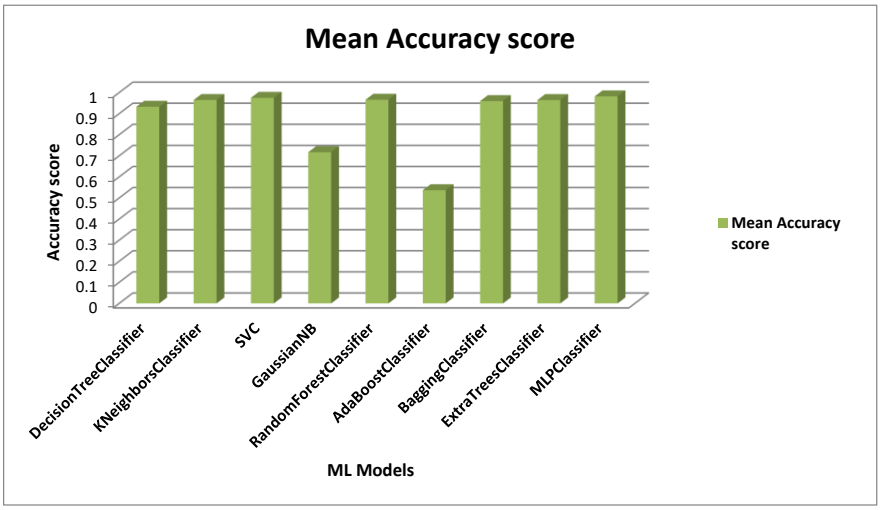

Fig. 3. Mean accuracy score

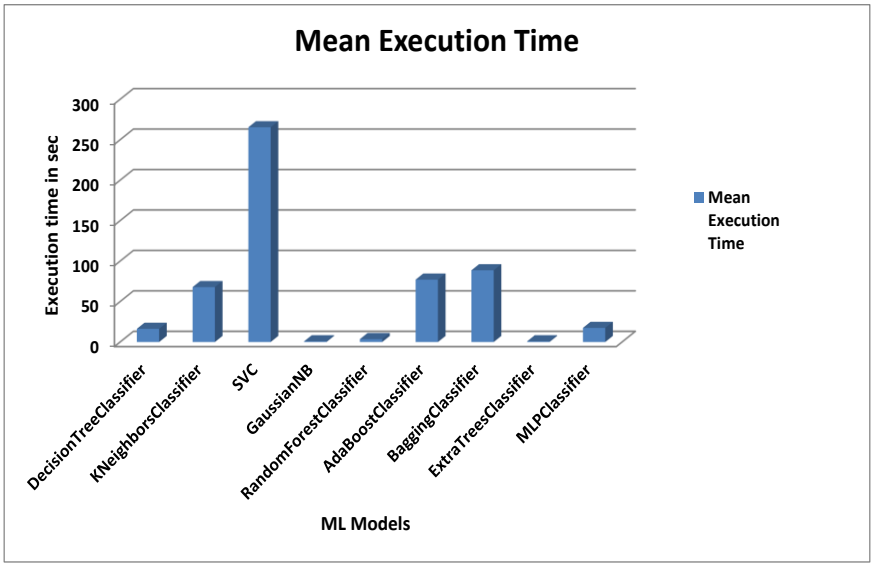

Fig. 4. Mean Execution Time

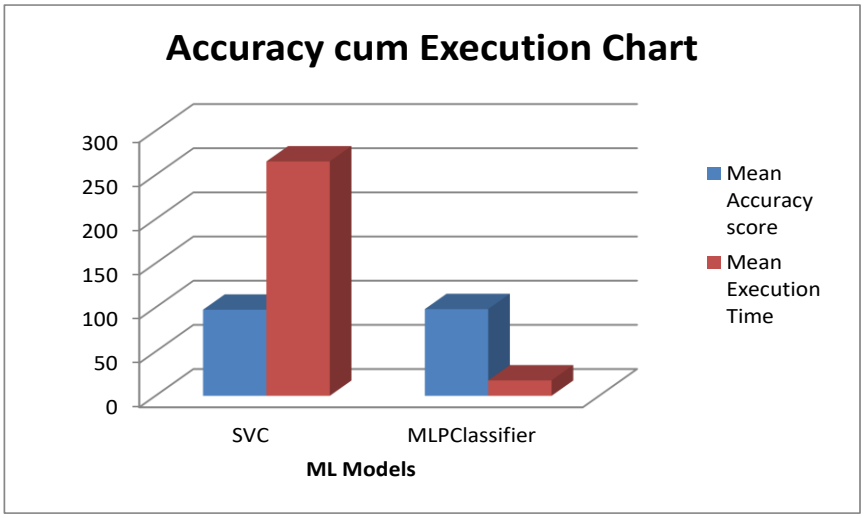

Fig. 5. Comparison Chart

E. Evaluation

By evaluating different ML model SVC and MLP both are giving good accuracy (table 2). Considering both the metric of accuracy and execution time, (fig. 5), it shows that MLP classifiers are best suited for HAR applications as they are providing good accuracy and taking less execution time. In fig. 5 X axis represents the two better performing models and y axis contains accuracy in percentage and execution time in sec. Support Vector Machine classifier can also be used provided its execution time is somehow reduced. Next prominent algorithms are KNN, Random Forest and Extra Tree classifiers.

Table- II: ML performance Table

Sr. No.	Parameters		
	Better performing models	Accuracy %	Execution time (sec)
1	SVC	97.5081	265.4305
2	ANN/MLP	98.187	17.45234

V. CONCLUSION AND FUTURE SCOPE

HAR is process of identifying activity. This activity could be fed as input to other monitoring or supporting applications. The sensor data, extracted features from data and classification models are main contributing elements in case of HAR. The classifiers plays important role in identification. This paper tries to find out the suitable models for HAR. We conclude that from the list of traditional supervised classifiers ANN/MLP performs better in case of HAR considering accuracy and execution time as its performance measuring criteria. After that SVM, kNN, Random Forest and Extra Tree are also probable candidates. In future we will try to enhance the accuracy or decrease the execution time of various classifiers which are performing well in case of HAR. New upcoming technology Deep learning can also be used in HAR for further improvements.

REFERENCES

1. I. Bhuvana and C. Yamini, “Survey On Classification Algorithms For Datamining : (Comparison And Evaluation),” Int. J. Adv. Res. Sci. Eng., vol. 4, pp. 125–134, 2015.
2. S. Chernbumroong, A. S. Atkins, H. Yu, and C. Mai, “Activity classification using a single wrist-worn accelerometer,” IEEE, 2011.
3. G. Kesavaraj and S. Sukumaran, “A Study On Classification Techniques in Data Mining,” in ICCCNT 2013, 2013.
4. S. K. Chaurasia, “Classification Techniques for Activity Recognition via Wearables,” in Computing for Sustainable Global Development, IEEE, 2018, pp. 4952–4957.
5. A. Tharwat, H. Mahdi, M. Elhoseny, and A. Ella, “Recognizing human activity in mobile crowdsensing environment using optimized k -NN algorithm,” Expert Syst. Appl., vol. 107, pp. 32–44, 2018.
6. S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, “Learning k for kNN Classification,” ACM Trans. Intell. Syst. Technol., vol. 8, no. 3, 2017.
7. Z. Zhang, “Introduction to machine learning : k-nearest neighbors,” Ann. Transl. Med., vol. 4, no. 11, pp. 1–7, 2016.
8. L. Demidova, E. Nikulchev, and Y. Sokolova, “The SVM Classifier Based on the Modified Particle Swarm Optimization,” Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 2, pp. 16–24, 2016.
9. D. Bzdok, M. Krzywinski, and N. Altman, “Machine learning : supervised methods,” Nat. Publ. Gr., vol. 15, no. 1, pp. 5–6, 2018.
10. D. Lara and M. A. Labrador, “A Survey on Human Activity Recognition using,” vol. 15, no. 3, pp. 1192–1209, 2013.
11. J. Han, M. Kamber, and J. Pei, Data Mining Concepts and Techniques, Third. Morgan Kaufmann, 2012.
12. M. Belgiu and L. Dra, “ISPRS Journal of Photogrammetry and Remote Sensing Random forest in remote sensing : A review of applications and future directions ˇ gut,” ISPRS J. Photogramm. Remote Sens., vol. 114, pp. 24–31, 2016.
13. L. V. Utkin and Y. A. Zhuk, “Robust boosting classification models with local sets of probability distributions,” Knowledge-Based Syst., vol. 61, pp. 59–75, 2014.
14. S. B. Kotsiantis, “Bagging and boosting variants for handling classifications problems: A survey,” Knowl. Eng. Rev., vol. 29, no. 1, pp. 78–100, 2014.
15. P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” Mach. Learn., vol. 63, no. 1, pp. 3–42, 2006.
16. C. F. Tsai, Y. F. Hsu, and D. C. Yen, “A comparative study of classifier ensembles for bankruptcy prediction,” Appl. Soft Comput. J., vol. 24, pp. 977–984, 2014.

AUTHORS PROFILE

Sunita Kumari Chaurasia has received her Bachelor in Engineering degree in Information Technology in 2003 (RGPV, India), completed Masters in Technology degree in Information Technology (USIT, GGSIPU, India) in 2010. She is pursuing her Ph.D in CSE at IGDTUW. Her area of research is context detection using machine learning. Currently she is working as Lecturer, Computer Engineering, DTTE, under Govt of NCT of Delhi.

SRN Reddy is working as Professor CSE Dept. and Dean Examination Affairs in Indira Gandhi Delhi Technical University for Women (IGDTUW), Delhi, INDIA. He has awarded PhD in the area of Embedded Systems Design in 2009 and M. Tech degree in 2002 from Jawaharlal Nehru University, New Delhi. His research interest includes Embedded Systems Design, Mobile Architecture and Programming, IOT, and Real-Time Systems. He has more than 15 years’ experience that includes research, industry and teaching.