Genetic Algorithm for Effective Fall Detection with Wrist Wearable Device

Abhilash Unnikrishnan, Abraham Sudharson Ponraj

Abstract: Falls have always been a major cause of injury related deaths among the old aged population in our country. It causes mental trauma and severe fractures to the bones and spine which impacts their quality of life. Therefore a proper fall prediction and alert system along with a timely rapid response could enable us to tackle such serious fall events and decrease the fatality. Various sensors and embedded controllers are used in conjunction with various machine learning classifiers to help us predict and optimize the falls effectively. This work presents a wrist wearable device using MPU-6050 sensor and raspberry-pi controller with help of machine learn algorithm which help us to predict the falls. Five different supervised learning algorithms and one unsupervised algorithm was implemented and evaluated on the basis of their accuracy, sensitivity and specificity. Out of all these classifiers, the decision tree with an accuracy of 85% was implemented in the system which classified the fall from the real time non-fall data sets. Further the performance of system was increased using genetic algorithm which gave better classification results unlike the normal decision tree classifier. Once the falls are predicted we can give a real-time response which can be an added feature to this system.

Keywords : Decision tree, Fall Detection, Genetic algorithm, Machine learning.

I. INTRODUCTION

Over the years there has been a huge demand in the health care and assistance for the elderly because of the steady increase in the old age population. Falls have become a major health risk faced by the old aged people which have led to many deaths. The World Health Organization has defined fall as an event which results in a person coming to rest inadvertently on the ground or floor or other lower level. They are a major health care problem which has to be handled effectively. Several thousands of fall related injuries has occurred each year which makes it the second most leading cause of deaths [1]. Therefore a proper detection and prediction of these falls along with timely medical assistance could help us avert big fractures and injuries.

A fall prediction system using the MPU6050 sensor with inbuilt Tri-axial accelerometer has been presented in the work, where the data from the sensor are collected and the necessary feature extractions are done to get the required features which is used to train the various classifiers. In this paper section II describes the literature survey, section III gives detailed information about the methodology and IV gives the detailed explanation of the classification algorithms. The later sections discuss about the genetic algorithm, hardware module and the conclusion.

Revised Manuscript Received on December 05, 2019

***** Correspondence Author
Abhilash Unnikrishnan, School of Electronics Engineering, VIT, Chennai, India. Email: abhy_1995@yahoo.com
Abraham Sudharson Ponraj*, School of Electronics Engineering, VIT, Chennai, India. Email: abrahamsudharson.ponraj@vit.ac.in

II. LITERATURE SURVEY

The major methods used for the fall detection are the Threshold and Machine Learning methods. A number of papers have been surveyed and it was observed that the accuracy of the machine learning methods were better when compared with the standard threshold based methods. Paper [2] describes a wrist wearable device using two threshold and machine learning algorithms. Later these algorithms are compared based on their accuracies. The machine learning approaches had the highest i.e. around 99% in the K-Nearest Neighbor (kNN) approach. Paper [3] describes a smart phone app and threshold based tri-axial fall detection system is described for effective fall detection and to make the data open for observation. When a fall occurs it directly sends SMS to the required center or the other contacts in the phone for timely assistance. In paper [4], a portable sensor based fall detection is suggested for an effective analysis. Here a tri-axial accelerometer, gyroscope, magnetometer and the smart phone is used from which the acceleration, yaw pitch and roll are found to get the position and orientation of the body. Falls and ADL's are simulated by placing these sensors at the waist, shoulder and foot of the subjects. Waist is considered as the best region for placing the sensors.

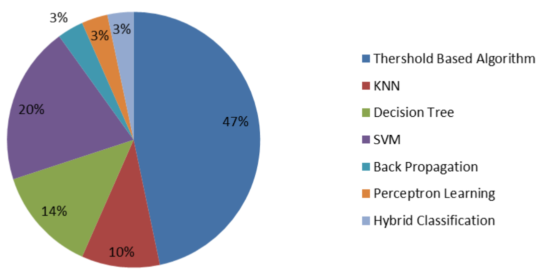

Fig. 1.Analysis of various fall detection algorithms

A kNN based machine learning algorithm based fall detection is described in paper [5]. The off shelf devices are used to develop this system. A video camera based surveillance system is presented which captures the silhouette of the person where the height and width of the image varies when the person is standing and in falling position.

The kNN classification algorithm is used to classify the postures using the ratio and difference of human body silhouette bounding box height and width. With the help of the kNN classifier and the critical time difference, a fall incident detection system is developed to detect fall incident events.

In paper [6] a waist mounted fall detection system is described. It compares both the threshold and machine learning algorithms and comes to a conclusion that the later has better accuracy but SVM had the best sensitivity and specificity. The main take away was that the serious consequence of falls among older adults is the ‘long lie’ experienced by individuals who are unable to get up and remain on the ground for an extended period of time after a fall. A weightless and wearable based body worn device is built in paper [7] using tri-axial accelerometer and gyroscope for simulating falls and ADLs. The fall events here were detected using a threshold based algorithm and using this falls could be distinguished from the ADLs.

A fall detection using five machine learning algorithms is described in paper [8] where the fall and non-fall activities are correctly classified using the Mobi-fall data sets. The kNN gave better accuracy and performance.

An optimized wireless and wearable sensor system using tri-axial accelerometer and gyroscope is discussed in this paper [9] which is placed at the chest to collect the real time data. It uses a threshold algorithm to detected and classify the falls.

The paper [10] presents an unobtrusive smartphone based fall detection system that uses a combination of information derived from machine learning classification applied in a state machine algorithm. The data from the smartphone built-in accelerometer is continuously screened when the phone is in the user’s belt or pocket. Upon the detection of a fall event, the user location is tracked and SMS and email notifications are sent to a set of contacts.

A novel fall detection algorithm based on one class support vector machine is described in paper [11]. The one-class Support Vector Machine (SVM) model is trained by the positive samples from the falls of younger volunteers and a dummy, and the outliers from the non-fall daily activities of younger and the elderly volunteers. This method can detect the falls effectively, and reduce the probability of being damaged in the experiments for the elderly people.

Paper [12] describes a fall detection system for the elderly and monitors them real time. The system has two components where one is a wearable device and the smart phone gives alerts to the emergency contacts for quicker assistance. If the acceleration exceeds a threshold, the device checks the gyroscope variation and alerted.

In paper [13] a fall monitoring system for old people using tri axial accelerometer is discussed where the device collects acceleration and the angle between elderly and horizontal plane of elderly people by MPU6050 tri-axial accelerometer, comparing the acceleration and angle that people and horizontal plane with threshold value to determine whether the old people fell.

In this paper [14], an effective fall detection algorithm for mobile platforms is proposed. Using data retrieved from wearable sensors, such as Inertial Measurements Units (IMUs) and/or Smart- Phones (SPs), the algorithm is able to detect falls using features extracted from accelerometer and gyroscope. Similarly a fall detection algorithm and a classification algorithm for activities of daily living using a wrist-worn wearable device are described in paper [15].

A waist worn fall detection system is discussed in this paper [16] and tri-axial accelerometer (ADXL345) was used to capture the movement signals of human body and detect events such as walking and falling to get a reasonable degree of accuracy. The algorithms like Multilayer Perceptron, Naive Bayes, and Decision tree, Support Vector Machine, ZeroR and OneR were used here for fall detections and to test their performance of classification. It is also proposed that multilayer perceptron has the better accuracy to classify the falls from the ADLs.

A hybrid classification algorithm is proposed in paper [17] to achieve accuracy in detection by tracking objects and having the ability to handle the causes. It helps to detect fall with high accuracy and reliability. The advantages of this method are that it has less computation complexity and improves efficiency of fall detection compared to existing machine learning algorithms. Fig.1 and Fig.2. give a detailed graphical analysis of the literature review where we find that waist is the suitable place for placing the center because of its closeness with the center of gravity but in this paper wrist was chosen because it is easy for analysis and the stigma for having a medical equipment attached to the body could be avoided.

III. METHODOLOGY

This work implements various machine learning algorithms onto the sensor data which has a mixture of several falls and non-fall activities when the person is walking on the flat ground. The flow of the work is shown in Fig. 3. and Fig. 4. which depicts the blocks of the fall detection system. The wearable sensor is worn on the wrist which is used for data acquisition. MPU 6050 is embedded in it along with the R-pi controller. This sensor has an inbuilt digital motion processor, tri-axial accelerometer and tri-axial gyroscope. This accelerometer measures the acceleration and the position of the body along the three dimensional axis and has programming range of ±2g to ±16g. The tri-axial gyroscope gives the angular acceleration on the 3D axis in deg/sec. It uses Inter-Integrated Communication (I2c) to transfer the sensor data to the controller. It is not a very expensive sensor.

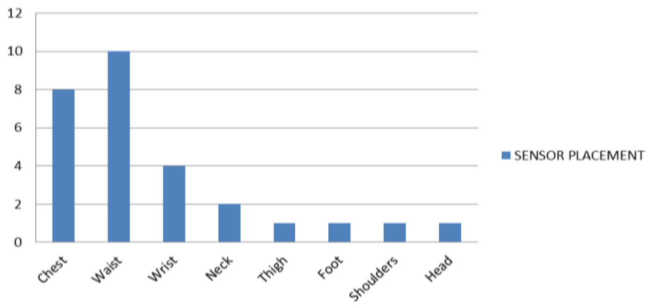

Fig. 2.Analysis of Sensor placement

For this fall detection system we consider the accelerometer data only [3]. Feature extraction is an important process in this fall prediction system. We have to build the derived values from the measured data. The sensor output from the accelerometer gives the Ax, Ay and Az i.e. the acceleration along each of the axis. From this we calculate the parameters like the total acceleration, vertical

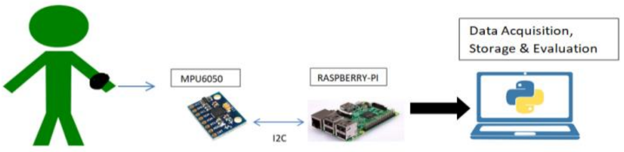

Fig. 3.System Outline

acceleration, and total velocity, vertical component of velocity, total displacement and vertical displacement respectively. We cannot process anything by considering only the individual axial accelerations therefore finding total magnitude of acceleration including the three axes is needed.

$$\text{Total Acceleration} = \sqrt{Ax^2 + Ay^2 + Az^2} \qquad (1)$$

Refer to (1), it shows the formula for calculating the Total acceleration. The features that are extracted from the sensor readings are shown in Fig. 5.

From total acceleration we get "Total velocity" by integrating it with respect to time. "Vertical acceleration" is obtained by considering the vertical components i.e. in the direction of gravity which is perpendicularly downwards. By integrating this vertical acceleration with time we get "Vertical Velocity". Similarly integration of the total velocity with time gives the "Total Displacement" and time window integration of vertical velocity gives "Vertical Displacement" [2].

In this fall detection system we have used the machine learning algorithms for classifying the falls from non-fall data. The supervised learning algorithms used are K-nearest neighbors, Logistic Regressions, Decision Trees, Support Vector Machines and Linear Discriminant Analysis. These data sets have to be labeled before introducing them to these classifiers. These fall and non falls from data are predicted by using the machine learning models and libraries under Python.

The steps for building the models for the classification are done by a step by step process:

- Loading Data - Reading the CSV data.
- Selecting the required Feature.
- Splitting the data using train_test_split (x, y,test_size=1/4). If 5000 data sets are present then 1 part of it is used for testing and the rest 3 parts for training to get better prediction.
- Model development & prediction - choose the required classifier, fit(x_train,y_train) and predict(y_test).
- Model evaluations using confusion matrix. [18]

IV. CLASSIFICATION ALGORITHMS

The various supervised and unsupervised classifiers used in this work are elaborated below.

K-Nearest Neighbors (kNN) is a lazy learning algorithm which plots all the features of the test data across an x-y plane. Later it classifies them as fall or non-fall. When a new feature vector is introduced it compares and calculates the Euclidean distance between the closest neighbors [19].

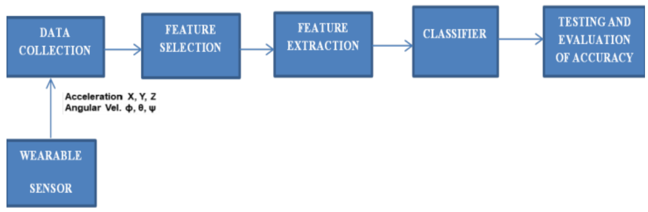

Fig. 4.Flow Chart

In Decision Trees (DT) algorithm the classifications are done based on a tree structure where each node is a variable and its parameters have the evaluation [20]. Where as in the Support Vector Machines (SVM) the features are taken to a hyperspace using nonlinear mapping where an optimum hyper plane is found separating two classes from a given data set. The hyper plane used to separate the two classes by creating a decision boundary.

Logistic Regressions (LR) is a supervised learning algorithm used for predicting the binary classes. It uses logistic function or the usual sigmoid function for predicting the output class. If the output of the sigmoid function is greater than infinity then the output predicted class is 1 or the fall. Else there is no fall at the intermediate.

K-Means is an unsupervised clustering algorithm. It doesn't require all the labeling of the data that is usually done in the supervised learning. The classifier itself labels the features into the respective categories. The clustering algorithm partitions n objects into k clusters where each object belongs to cluster of nearest mean.

These classifiers are trained and later compared with the Test data. If they are compared and the values are close the accuracy of the classifier is 100% which is the ideal case.

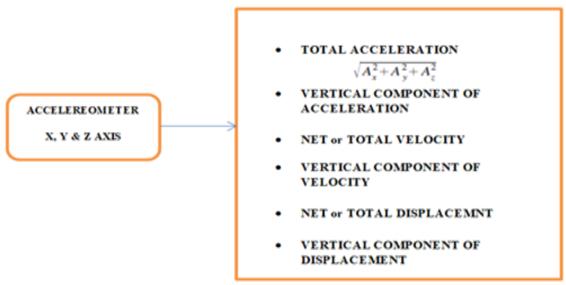

Fig. 5.Features Extracted

From Table-I of the classifier results we can infer that Decision Tree algorithm gives better classification of the fall from the non-fall data. It has better accuracy (fraction of correct predictions made) from the other classifiers.

To evaluate the performance of the classifier and to analyze the correct and incorrect predictions made visually we use the confusion matrix plots.

It is one of the key tools in analyzing the machine learning classifiers. From this we could even measure the performance parameters of the classifier like the sensitivity and specificity. Using this confusion matrix the numbers of correct and incorrect predictions are summed up class-wise. The Diagonal value of the matrix represents the accurate predictions, while non-diagonal elements are inaccurate predictions which are show in Fig. 6.

Table- I: Classifier Results

	kNN	SVM	DT	LR	K-Means & SVR
Accuracy (In %)	84	82.6	85	82.6	66.96
Sensitivity (In %)	36.95	0	64.13	0	0
Specificity (In %)	94.96	100	89	100	100

- True Positive (TP), fall has occurred and the system has detected the fall.
- False Positive (FP), fall has not occurred but the system has detected a fall.
- True Negative (TN), No fall occurred and the system also detected the no fall.
- False Negative (FN), Fall Occurred but the system did not detect the fall.

Sensitivity is the capability of identifying all the true positives (falls) and specificity is the capability of identifying all the true negatives (no falls). From the classifier results we can infer that Decision tree has comparable values of both the sensitivity and specificity that is needed for proper fall prediction. The confusion matrix plot for the decision tree classifier is shown in Fig. 7.

$$Sensitivity = \frac{TP}{TP+FN} \qquad (2)$$

$$Specificity = \frac{TN}{TN+FP} \qquad (3)$$

The equations for sensitivity and specificity are calculated as shown in (2) and (3).

V. GENETIC ALGORITHM

To improve the classification accuracy of the decision Tree classifier we go for the performance enhancement algorithm which is the genetic algorithm. Here this algorithm is used for feature selection because it helps us to select only those best features from the entire feature set that contributes for the better accuracy of the classifier. Thus it helps us to remove the unnecessary features that decrease this classifier accuracy. The algorithm creates lots of combinations of the features to check which combination gives the better accuracy which adds more work on the entire system, if the input features to this algorithm are very large. [18]

The Genetic Algorithm is a heuristic optimization method inspired by that procedure of natural evolution. In feature selection, the function to optimize is the generalization performance of a predictive model.

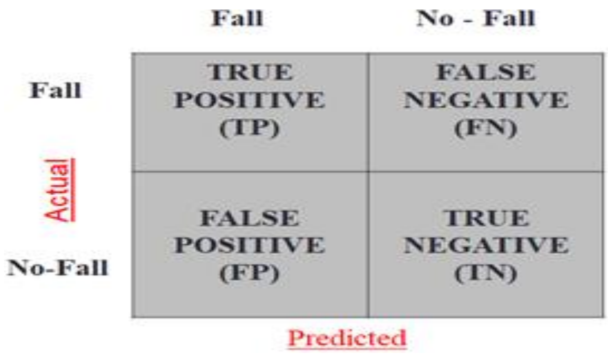

Fig. 6.Confusion Matrix

More specifically, we want to minimize the error of the model on an independent data set not used to create the model [19]. They operate on the population to give better outputs. For every generation, new population is created by selecting those individuals based on the fitness function and later combining them together using operators from genetics. This produces offspring's and can also undergo mutation. From these newer populations are produced based on evaluation of those individuals that can adapt better.

Each individual in the population represents a model. The number of genes is the total number of features in the data set. Genes here are binary values, and represent the inclusion or not of particular features in the model. The number of individuals or population size must be chosen for each application. [20].

The entire flow of the genetic algorithm can be described as:-

- Initialize the population.
- Assign the fitness values.
- Selection process chooses those individuals that can combine for next population.
- Cross over recombines the above step 3 selected individual to generate a newer population.
- The mutation step is done next if the results after cross over produces offspring's that are similar to the parent who causes low diversity to the population. The values of these offspring's are changed at random by the mutation.
- Now we have a new population and the step 2 repeats all over if criteria is false else if better output is achieved then the algorithm stops.

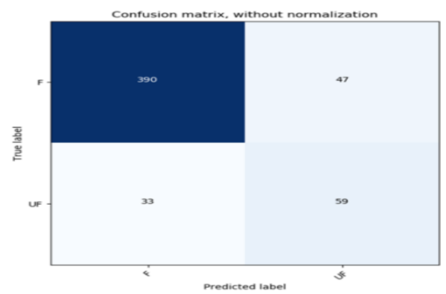

Fig. 7.Confusion Matrix Plot For Decision Tree Classifier

From Table-I we can infer that the Decision Tree classifier has over 85% accuracy. With the help of the genetic algorithm we can boost this accuracy to 100 percent. This result is shown in Table-II with a population of 2115 and for about 10 generations [21].

Table- II: Genetic Algorithm Results

Generations	Netvals	Average	Min	Max
0	2115	0.151	0	1
1	1302	0	0	1
2	1260	0.9874	0.398	1
3	1253	0.9952	0	1
4	1263	0.995	0.399	1
5	1269	0.9964	0.399	1
6	1254	0.996	0.399	1
7	1275	0.9963	0.72851	1
8	1271	0.9967	0.72851	1
9	1301	0.996	0	1
10	1272	0.995	0	1
Best Accuracy achieved : 100 %				
Number of features considered is 4				
Individuals : [1, 1, 0, 1, 0, 1]				
Feature Subset : {'TA', 'VA', 'VV', 'VD'}				
Accuracy of this feature Subset : 100%				

The algorithm has chosen the 4 (out of the 6 features given as inputs) features that contribute to the 100% accuracy. For the 10 generations it calculates the best feature which gives us the better results. The Table-III shows the comparative results of the various errors calculated for the Decision Tree algorithm and the Decision tree with had the Genetic algorithm. From the results we can infer that better accuracy and lower errors have been achieved while using the performance enhancement algorithm.

VI. HARDWARE MODULE

Entire fall detection system set up is shown in Fig 8. The IMU sensor has been interfaced to the Raspberry pi board. A USB cable connects the Raspberry pi to the laptop. Python programming language is used in this system to communicate with GPIO Ports. Using this wearable device module readings are taken for fall and non-fall cases on a normal walking plane for a period of every 20 seconds.The hardware module is worn on the wrist and the readings are taken for a specific period of time intervals. These readings from the sensor are converted into a .csv format with labeling before uploading it to the classifiers.

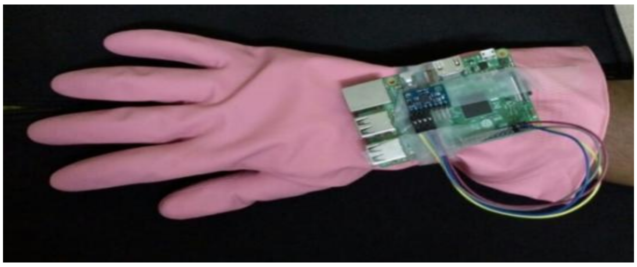

Fig. 8.The Hardware Setup

Table- III: Error Calculation Results

Algorithms Used	Mean Absolute Errors	Mean Sq Errors	Root Mean Sq Errors	R^2 Errors
Decision Tree	0.15122	0.151	0.388	-0.0526
Decision Tree & GA	0	0	0	1

VII. CONCLUSION

To help the elderly patients in the health care sector a wrist wearable sensor device could enable the doctors to get suitable alerts and provide timely health assistance to the old patients at a proper time which helps to avoid serious injuries or even death. The wrist is the best comfortable position for having the system because of its easy wearable nature and comfortable removal. The machine learning based fall prediction system could classify the fall data form the regular Activities of Daily Living. The data from the Inertial Measurement Sensors are also available on the systems which could be made available to the patient's family or doctors as a live monitoring system.

ACKNOWLEDGMENT

We like to thank Vellore Institute of Technology, Chennai management for their moral support. We would also thank the professors of our university for their guidance and valuable suggestions.

REFERENCES

1. https://www.who.int/news-room/fact-sheets/detail/falls What the Internet of Things (IoT) Needs to Become a Reality by Freescale
2. de Quadros, Thiago, Andre Eugenio Lazzaretti, and Fábio Kürt Schneider. "A movement decomposition and machine learning-based fall detection system using wrist wearable device." IEEE Sensors Journal 18.12 (2018): 5082-5089.
3. Sposaro, Frank, and Gary Tyson. "iFall: an Android application for fall monitoring and response." Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE. IEEE, 2009.
4. Mao, Aihua, et al. "Highly portable, sensor-based system for human fall monitoring." Sensors 17.9 (2017): 2096.
5. Liu, Chien-Liang, Chia-Hoang Lee, and Ping-Min Lin. "A fall detection system using k-nearest neighbor classifier." Expert systems with applications 37.10 (2010): 7174-7181.
6. Aziz, Omar, et al. "A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials." Medical & biological engineering & computing 55.1 (2017): 45-55.

7. Guo, Han Wen, et al. "A threshold-based algorithm of fall detection using a wearable device with tri-axial accelerometer and gyroscope." *Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2015 International Conference on*. IEEE, 2015.
8. Vallabh, Pranesh, et al. "Fall detection using machine learning algorithms." SoftCOM. 2016.
9. Huynh, Quoc T., et al. "Optimization of an accelerometer and gyroscope-based fall detection algorithm." Journal of Sensors 2015 (2015).
10. Aguiar, Bruno, et al. "Accelerometer-based fall detection for smartphones." Medical Measurements and Applications (MeMeA), 2014 IEEE International Symposium on. IEEE, 2014.
11. Zhang, Tong, et al. "Fall detection by wearable sensor and one-class SVM algorithm." Intelligent computing in signal processing and pattern recognition. Springer, Berlin, Heidelberg, 2006. 858-863.
12. Santiago, Joseph, et al. "Fall detection system for the elderly." Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual. IEEE, 2017.
13. Luo, Jing, Bocheng Zhong, and Dinghao Lv. "Fall Monitoring Device for Old People based on Tri-Axial Accelerometer." IJACSA) Int J Adv Comput Sci Appl 6.5 (2015).
14. De Cillis, Francesca, et al. "Fall-detection solution for mobile platforms using accelerometer and gyroscope data." Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE. IEEE, 2015.
15. Yuan, Jian, et al. "Power-efficient interrupt-driven algorithms for fall detection and classification of activities of daily living." IEEE Sensors Journal 15.3 (2015): 1377-1387.
16. Kerdegari, Hamideh, et al. "Evaluation of fall detection classification approaches." Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on. Vol. 1. IEEE, 2012.
17. Sheryl Oliver A et Al. Optimized Low Computational Algorithm For Elderly Fall Detection Based On Machine Learning Techniques.
18. https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection
19. https://scikit-learn.org/stable/
20. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/
21. https://github.com/renatoosousa/GeneticAlgorithmForFeatureSelection.

AUTHORS PROFILE

Abhilash Unnikrishnan received the graduate degree in Electrical and Electronics Engineering from Anna University in 2016, M.Tech in Embedded Systems from Vellore Institute of Technology, Chennai in 2019. He is currently a working professional, pursuing his career in the automotive embedded domain. He has published one journal paper. His areas of interest are in core embedded systems, automotive electronics, Internet of Things and machine Learning.

Abraham Sudharson Ponraj received the graduate degree in Electronics and Instrumentation Engineering from Karunya University in 2010, M.Tech (Embedded Systems) from Kalasalingam University 2012. He is currently a PHD student and working as an Assistant Professor at Vellore Institute of Technology, Chennai, India. He has more than 6 years of teaching experience and has published more than 10 journal papers. He is an IEEE members and his area of interest is embedded system and IoT for sustainable agriculture and farm management.