

# CURA: Real Time Artificial Intelligence and IoT based Fall Detection Systems for patients suffering from Dementia

Sanket Mishra<sup>1</sup>, Bernard Ngangbam<sup>2</sup>, Shritik Raj<sup>3</sup>, Nihar Ranjan Pradhan<sup>4</sup>

<sup>1,2,3,4</sup>School of Computer Science & Engineering, VIT-AP University, Amaravati, 522237, Andhra Pradesh, India

## Abstract

According to the rising concern of the effects on the families due to dementia suffering patients, we aim to provide caretakers a work-life balance in which monitoring can be done with much more ease and efficiency in real time. This device can also be used in old age homes as well as hospitals which reduces the workload of the caretakers and helps them to easily monitor the patients. We aim to contribute for the betterment of the society and provide a virtual assistance for the patients suffering from dementia. The number of elderly people living alone has been increasing all over the world. If dementia has been detected at an early stage, the progress of disease can be slowed. The patients suffering from dementia are prone to falling quite frequently so as to detect that and to alert their caretakers to take necessary actions. In this study, we proposed a system in which we detect the real time state of the elderly people living alone by using the Machine Learning and IoT (Internet of Things) technology. We installed sensors inside a finger strap which is attached to the person. These sensors can detect the motions of the patient and predict their real time state to have a 24 by 7 support to provide assistance to the patients.

Received on 12 July 2023; accepted on 01 September 2023; published on 25 September 2023

**Keywords:** Fall Detection, Accelerometer, Gyroscope, Sliding Window, Timeframe, Classification, Alert, False Alarms

Copyright © 2023 S. Mishra *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi:10.4108/eetpht.9.3967

## 1. Introduction

In recent years, the world has been suffering from aging. There have been many studies estimating the number of elderly people living alone is increasing frighteningly. Moreover, patients who suffered from dementia also would be increasing. In 2025, we estimated the number of elderly people with dementia would be increased to 7 million.

Dementia is a disorder of memory and judgment caused by dying brain cells. Therefore, patients suffering from dementia are unable to live an ordinary social life. There is severe need to have a device which could monitor the actions of such patients as to provide aid as soon as possible and to reduce the burden of the caretakers as much as possible.

The need for flexible electronic devices that can be comfortably worn by our target subject, i.e., elderly

people suffering from Dementia needs to tick the elements of the device being lightweight, wearable, reliable, non-invasive and skin comforting. A recent study [1] in 2018, describes potential challenges of developing such wearable healthcare devices.

We aim to create a device that helps the family members and the doctors of the elderly patients suffering from dementia to be alerted in real time about their state with the help of the Arduino pro chip named, Nicla SENSE ME. The chip has a BHI260AP motion sensor system with integrated AI, BMM150 magnetometer, BMP390 pressure sensor, and BME688 four in one gas sensor with AI integrated high linearity, as well as high accuracy pressure, humidity, and temperature sensors. The use of wearable device (i.e., Finger Splint) in order to monitor the day-to-day activities of a person for the health care is quite of importance. The use of sensors like accelerometer and gyroscope plays a vital role in this regard. For this purpose, human activities are divided into two

\*Corresponding author. Email: [sanketmishra@live.com](mailto:sanketmishra@live.com)

categories of static postures like standing, bending, sitting, and lying and dynamic transitions are the movements and motions between the different static postures. The accelerometers are used to detect static postures and gyroscopes are used to detect dynamic transitions. We propose CURA (pronounced 'koo-rah' is the Latin word for 'Cure'), a real-time fall detection system for patients suffering from dementia

The main contribution of this paper is to present a unified solution for fall detection and emergency alert. The device can monitor the patient's activities at home and unfamiliar surroundings as compared to the solutions that are limited to certain boundaries.

## 2. Related Work

In context of a fast aging population the use of advanced technology driven smart wearables need to be comfortably accepted by the subject in order to target creating such devices. J. Li, Q. Ma, A. Chan and S. S. Man et al. [2] took a survey of 146 samples from elderly people over the age of 60 years to gain insights on the acceptance and preference of older adults to practice use of such devices.

Another similar study where D. Dias and J. P. Cunha et al. [3] reviewed important aspects of these wearable devices focusing on vital monitoring systems listing out the state-of-art of it. They also discussed on extending this to smart clothing wear for medical purposes and finally targeted to learn the market trends and the future challenges of such devices.

There has been a lack of dataset for creating health care devices such as fall-detection models using sensors. E. Casilari, J. Santoyo and J. M. Garcia et al. [4] proposed a new dataset UMAFall where they acquired movement traces using multisensors, including the sensors used in this paper accelerometer and gyroscope, which is used in any movement tracing related research.

Another work similar to CURA was proposed a long time back in 2006, but highly relevant as a benchmark as C. Lin, M. Chiu, C. Hsiao, R. Lee and Y. Tsai et al. [5] proposed global positioning system (GPS) into healthcare related wearable devices, which is crucial in case of key events occurring. They provided for family members and volunteer caretakers to be able to identify the real-time positions of missing elderly people.

Recently, in 2019 M. Al-khafajiy, T. Baker, C. Chalmers, M. Asim, H. Kolivand, M. Fahim, and A. Waraich et al. [6] proposed a remote health monitoring system to monitor vitals via a mobile app portal using Arduino UNO with alert sending mechanism to hospitals using cloud services transferring data from homes to hospitals through a cloud datacenter. The proposed Smart Healthcare Monitoring System (SMHS) used smart phones and wearable sensors to monitor elderly people in real-time, with the functionality of

being able to conveniently monitor remotely at home and alert the hospital unit in case of any emergency seamlessly via the cloud service.

Another related work published in 2018 by H. Nweke, T. Wah, M. Al-Garadi and U. Alo et al. [7] where they performed feature extraction in order to reduce computation complexities for the proposed deep learning algorithms. We also adapted a similar approach to reduce the computational complexity using feature extraction using ts-fresh as described in (3.4).

## 3. Methodology

### 3.1. Setup Architecture

The architecture that we have setup for creating an smart healthcare system for fall-detection, proposed in this work is CURA (pronounced as 'koo-rah', meaning 'Cure' in Latin) as shown in Fig 1.

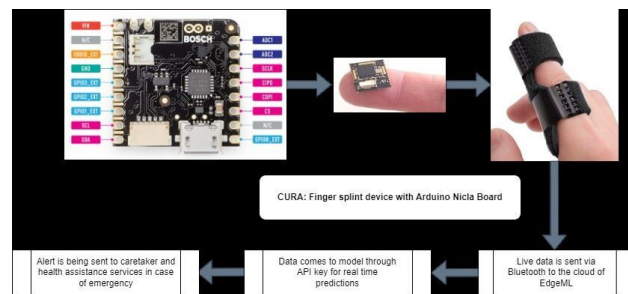


Figure 1. A visual representation of the setup of CURA

### 3.2. Data Collection

First, we gather the required data for through the available sensors present on Nicla SENSE ME board i.e., accelerometer (x, y, z) and gyrometer (x, y, z) for detecting the movement pattern of the subject for creating TinyML [8] models. We collect data simulating the movements of a patient suffering from dementia and by recording the possible cases that we need to detect, in this case especially focusing on fall/not fall detection mechanism and walk/not walk detection mechanism. A group of around 10 college students helped us in simulating the main events (sit/sleep/walk/fall) by collecting data for 25 days at irregular intervals throughout the day by trying to simulate the events as close as possible to our target subject suffering from dementia (slow and small-scale movements). Fig 2. and Fig 3. represents visualisation of one such data instance recording of Fall and Walk respectively. Similarly, we collected several test cases within a span of 7 days, in small instances keeping in mind that it should represent the simulation of our target subject a dementia patient.

Mainly two sensors for used for the purpose for CURA, namely gyroscope and accelerometer that is

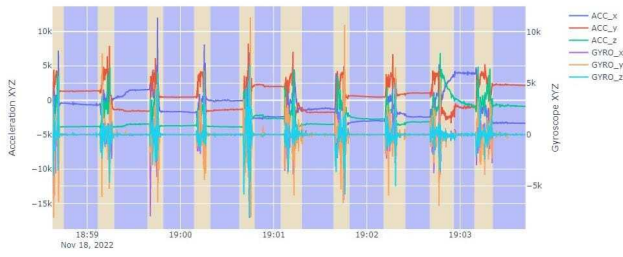


Figure 2. Data visualization of Fall/Not-Fall signal instances collected

embedded in the Arduino Board Nicla SENSE ME. The Nicla SENSE ME Board was attached on a finger splint, as shown in Fig . The subjects who contributed for the data collection were all batch mates whose age range from 19-22 years.

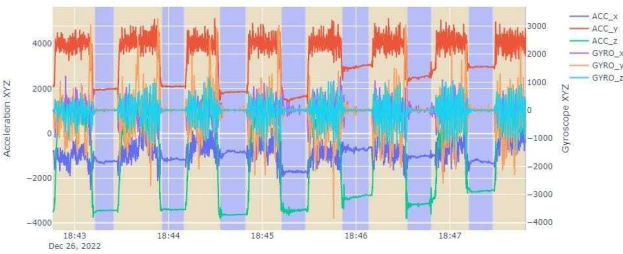


Figure 3. Data visualization of Walk/Not-Walk signal instances collected

Due, to limited resources and computational power of our localhost systems, we took a comparatively low sampling rate of 15 Hz for each sensor.

As we are dealing with fall-detection we labelled the recorded data instances into Fall class, indicating fall cases and Other class, indicating other events like Walk, Sleep, Sit, Stand, etc. Our collected data comprises of a total 33,858 data samples where 7,708 instances for Fall class, and 26,150 instances for Other class which included, 7,565 instances for sitting, 7,292 instances for sleeping, 7,572 instances for walk, and 3,721 instances for climbing stairs.

### 3.3. Data Labelling

The data collected is just raw values of x, y, z axes readings at a given sample rate. Using EdgeML [9] we label the data as shown in Fig 4. into each class i.e., Fall class and Other class (Sit/Sleep/Walk/Climbing Stairs) to be able to visualize and train the pattern that exists from the readings. These set of labelled data are then stored in the EdgeML [10] database and can be accessed for training machine learning models to detect fall for CURA.

Data labelling is done such that we label data points into Fall class and Other class inorder to reduce

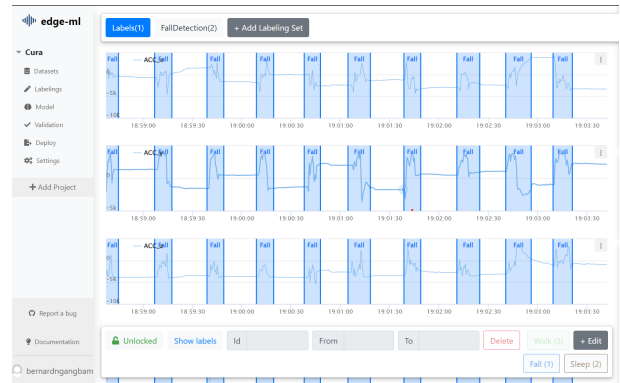


Figure 4. Labelling of Fall signal instances in EdgeML

complexities of creating different machine learning models for other use cases too such as Walk, Sit, Sleep, Climbing, etc combined together which is not feasible with limited computation power of our on-site localhost computers used for this proposed work and thus be able to focus on the target of fall detection using edge computation [11]. However, for flexibility purposes of CURA we have targeted two major classification, i.e., Fall/Not-Fall classification and Walk/Not-Walk classification as described in (3.5).

### 3.4. Feature Extraction and Sliding Window Size

Initially, the data being collected is a time series data in which the data points are plain readings from the sensors of the board at a sampling rate of 15 Hz for each sensor enabled.

Since, we have the data points in such a way that they are just the x, y and z values of accelerometer and gyro meter not meaning anything, we add a window size to define a certain action/pattern within a timeframe as required. This is based on the pattern of the readings at a certain window timeframe, so now the model can learn the pattern/action rather than just the raw values which would mean null. So, say for a window size of  $k = 10$ , we will have the data points of the time series from  $t = 1/15 * 10$  s in a single sliding window.

Now, in order optimize this further we perform feature extraction using ts-fresh [12] on each and every sliding window to extract the features contributing crucially to the classes Fall/Not-Fall or Walk/Not-Walk. Upon feature extraction using ts-fresh we get new features for each channel i.e., for acc\_x we now have the following,

- 'acc\_x\_sum\_values',
- 'acc\_x\_median',
- 'acc\_x\_mean',
- 'acc\_x\_length',
- 'acc\_x\_standard\_deviation',
- 'acc\_x\_variance',
- 'acc\_x\_root\_mean\_square',

'acc\_x\_maximum',  
 'acc\_x\_absolute\_maximum',  
 'acc\_x\_minimum'

For, acc\_x we get 10 such extracted features, that is the inferred values from that particular sliding window [13]. Similarly, we get 60 such total extracted features for each timeframe for 6 sensor values (acc\_x, acc\_y, acc\_z, gyro\_x, gyro\_y, gyro\_z).

### 3.5. Target Classification

Since, the data collection is only limited to a very small sample rate and very less battery span using very less computational power we have only two major working classifications. The Fall/No-Fall classification works such that in case the subject is remaining seated or lying down for a rest, or is simply walking around at their normal pace its target is to classify such patterns as No-Fall. As soon as there is an accident and a loss of muscle-memory coordination resulting in the falling of the subject, the target is to classify a Fall scenario and alert the caretakers.

Similarly, the Walk/No-Walk model works such that the target for the pattern in which the subject is doing anything other than walking is No-Walk. As soon as the subject starts, walking around, climbing the stairs the target then is Walk which can also be monitored simultaneously.

### 3.6. Possible Cases

**Sitting/Sleeping:** In this case, the targets expected are No-Walk and No-Fall.

**Standing Still:** In this case, the targets expected are No-Walk and No-Fall.

**Walking:** In this case, the targets expected are Walk and No-Fall

**Falling:** In this case, the targets expected are Walk and Fall.

Here, we have Walk then fall as the fall is a result of muscle-memory coordination loss resulting to go from walk phase to a fall. So, generally we can have three cases:

a) **Safe/static:** Sitting/Sleeping/Standing does not require much attention from the caretakers as the subject and most particularly safe.

b) **Movement/Warning:** In this case the subject is walking around and is not advised to do so in case of no supervision and needs to be supervised immediately to prevent any accident.

c) **Falling/Alert:** Here, the subject has most likely undergone a fall or something that needs to be immediately looked upon by the caretaker as soon as possible.

The above test cases were all simulated and recorded on with variances and permutations trying to imitate the action of a probable subject as close as possible.

## 4. Model Creation

### 4.1. Timeframe labelling for learning pattern

As we are dealing here with a timeseries data, simply training a model to learn the possible label i.e., Walk, No-Walk, Fall or No-Fall from the features, x y and z readings of accelerometer and gyro meter would be meaningless as there is no relation with the action done before and the action coming after the present one. So, to get the pattern which is to be learned by the model for the possible labels we need to be able to relate the data points of the timeseries data.

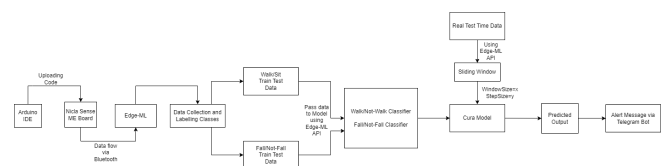


Figure 5. Model Architecture

We obtain this by using a sliding window. As shown below, a sliding window now collects the datapoints of a particular timeframe and each has its own label i.e., the action that is recorded at that particular timeframe (Fall/No-Fall or Walk/No-Walk cases). If we have datapoints for an instance of the subject walking for 15 secs and then sitting down for the next 15 s, and we have a sliding window of size 5 and a step size of 5, we will have the following:

- Timeframe 1: Walk (0-4)
- Timeframe 2: Walk (5-9)
- Timeframe 3: Walk (10-14)
- Timeframe 4: No-Walk (15-19)
- Timeframe 5: No-Walk (20-24)
- Timeframe 6: No-Walk (25-29)

So, for a sliding window of size k, and a step size of m, and n samples at a certain sample rate, for example we have,

```

0 1 2 3 4 5 6 7 8 9
0 0 0 0 0
_ 0 0 0 0 0
__ 0 0 0 0 0
___ 0 0 0 0 0
____ 0 0 0 0 0
_____ 0 0 0 0 0
    
```

Here, we have n = 10 samples, window size k = 5 and step size m = 1, therefore number of timeframes = 6. Now, as we have a pattern (at a timeframe) and



the particular label for the action being recorded at that instance we can now train the model using different algorithms for classifying the labels we need for this purpose.

#### 4.2. Parallel Walk/No-Walk and Fall/No-Fall models

Now, we have a two parallel model working architecture that predicts the two main cases: Walk/No-Walk and Fall/No-Fall, which leads us unto our 3 main possible cases discussed earlier.

- a) **Safe/Static:** 0/0 (No-Walk/No-Fall)
- b) **Movement/Warning:** 1/0 (Walk/No-Fall)
- c) **Falling/Alert:** 1/1 (Walk/Fall)

Depending on the output of the setup architecture we obtain our required target predictor for CURA by combining the output of the two base models. For learning the patterns, we used 6 different classifier algorithms, namely Logistic Regression, Random Forest, Decision Tree, Gaussian Naive Bayes, Support Vector and K Neighbors Classifier.

As for learning each pattern, we require those important features from the original raw data points of the reading values of the used accelerometer and gyro meter we use ts-fresh for extracting the features of the timeseries data.

The problem here is that in the real-world scenario when an emergency event has occurred, and then having to run all 6 models parallelly is computationally expensive and is not the best suited approach when time is a key factor in saving the subject's life or providing the needful attention immediately.

#### 4.3. Optimal Model Selection Algorithm

We trained several models using different 6 machine learning [14] [15] algorithms namely, Logistic Regression [16], Random Forest [17], Decision Tree [18], Gaussian Naive Bayes, Support Vector, and K Neighbors Classifier [19]. So, we obtain different accuracies for each learning algorithm. Using the Optimal Model Selection Algorithm approach, we select the most optimal learning approach for specific test cases, thereby optimising on time and having to classify using only the best optimal learning path rather than having to process all the 6 models parallelly. This makes the work set for the real-world challenges.

### 5. Results and Discussions

In this research, we achieve the best optimal learning algorithm r-squared score using Decision Tree classifier at a sliding window of size  $k = 4$ , which gives a high accuracy score of 95% using the r-squared metric. Other, classifiers such as Naive Bayes, Random Forest and K-Neighbors classifiers also give an accuracy of

above 90% but however, Decision Tree works the best as it gives the output in lesser time, which is a key factor here as dealing with emergency cases, time is key.

Given, below Table 1. shows the accuracy (r-squared) values of the 6 different classifiers at window size ( $k=2,4,6$ ).

**Table 1.** Accuracy scores (r-squared) on different window size (k) values

Machine Learning Models	Window Size k values		
	k=2	k=4	k=6
Logistic Regression	0.85	0.90	0.89
Random Forest	0.90	0.90	0.94
Decision Tree	0.70	0.95	0.68
Support Vector	0.90	0.95	0.94
Naive Bayes	0.85	0.85	0.94
K Neighbors	0.90	0.95	0.90

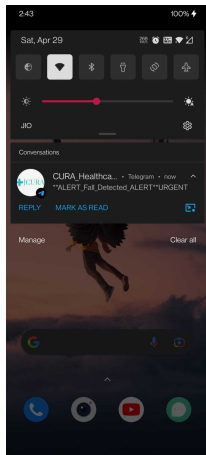
Since, the data was collected by simulation of an elderly person suffering from dementia and not with an actual subject there are some limitations that comes with the simulation. Other limitations such as lower battery power of the device, limited bluetooth range of the Arduino Board, limited storage in the EdgeML database, not being able to record a single data instance for any longer than around 30 minutes (there were multiple takes recorded to increase data samples for learning), and low sampling rate of the sensor i.e., 15 Hz and low memory resources available for collecting larger amount of data and simulation of the possible cases mentioned (3.6).

Optimising and eliminating these limitations in future research works the model accuracy scores and the overall working can be improved.

### 6. Conclusion and Future Work

In this research we have used cutting edge technology to develop a fall-detection wearable in from of a finger splint by using IoT sensors embedded in the Arduino Nicla SENSE ME board which works on edge impulse and provides real time fall detection alerts to the caretakers and health assistance. The finger splint device is connected via bluetooth which transfers the data through the EdgeML cloud, which is sent to the model through API keys. The wearable is made by considering ergonomics, adjustable fit, durability and battery life as to give the subject a safe experience. The results for the Decision Tree model is the most optimal solution with an accuracy of 95% which is the highest compared to other models mentioned in (4.3). The alert message is sent to the caretaker and registered health assistance through as shown in Fig 6. using the chat bot

developed on Telegram to handle the emergency as soon as possible.



**Figure 6.** Alert Message via Telegram Bot to registered caretakers

In the future we also aim to consider corner cases such as false alarms, i.e., the subject dropping the finger splint device which could result in the case of a false alarm of a fall detection being triggered. For this case in future researches we aim to add a ground truth validation mechanism using a camera module attached that would capture snapshots in case of an event and cross verify it using object detection models or region based convolutional models (R-CNNs).

## References

- [1] HA, MINJEONG and LIM, SEONGDONG and KO, HYUNHYUB and DALY P.W. (2018) *Wearable and flexible sensors for user-interactive health-monitoring devices* (Journal of Materials Chemistry B), 6th vol.
- [2] LI, JUNDE and MA, QI and CHAN, ALAN and MAN, SIU SHING (2019) *Health monitoring through wearable technologies for older adults: Smart wearables acceptance model* (Applied Ergonomics), [10.1016/j.apergo.2018.10.006](https://doi.org/10.1016/j.apergo.2018.10.006)
- [3] DIAS, DUARTE and CUNHA, JOÃO PAULO (2018) *Wearable Health Devices—Vital Sign Monitoring, Systems and Technologies* (Sensors), 18th vol [10.3390/s18082414](https://doi.org/10.3390/s18082414)
- [4] CASILARI-PÉREZ, EDUARDO and SANTOYO RAMÓN, JOSÉ and CANO-GARCIA, JOSE MANUEL (2017) *UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection* (Procedia Computer Science), 110th vol [10.1016/j.procs.2017.06.110](https://doi.org/10.1016/j.procs.2017.06.110)
- [5] LIN, CHUNG-CHIH and CHIU, MING-JANG and HSIAO, C.-C and LEE, REN-GUEY and TSAI, YUH-SHOW (2006) *Wireless Health Care Service System for Elderly With Dementia* (Information Technology in Biomedicine, IEEE Transactions), 10th vol
- [6] AL-KHAFAJI, MOHAMMED and BAKER, THAR and CHALMERS, CARL and ASIM, MUHAMMAD and KOLIVAND, HOSHANG and FAHIM, MUHAMMAD and WARAICH, ATIF (2019) *Remote health monitoring of elderly through wearable sensors* (Multimedia Tools and Applications)
- [7] NWEKE, HENRY and WAH, TEH and AL-GARADI, MOHAMMED and ALO, UZOMA (2018) *Deep Learning Algorithms for Human Activity Recognition using Mobile and Wearable Sensor Networks: State of the Art and Research Challenges* (Expert Systems with Applications), 18th vol [10.1016/j.eswa.2018.03.056](https://doi.org/10.1016/j.eswa.2018.03.056)
- [8] KALLIMANI, RAKHEE and PAI, KRISHNA and RAGHUWANSHI, PRASOON and IYER, SRIDHAR and ALCARAZ LÓPEZ, ONEL (2023) *TinyML: Tools, Applications, Challenges, and Future Research Directions* (Sensors)
- [9] *EdgeML - An open source and browser based toolchain for machine learning on microcontrollers* <https://edge-ml.org/>
- [10] PINYARASH PINYOANUNTAPONG and PRABHU JANAKARAJ and RAVIKUMAR BALAKRISHNAN and MINWOO LEE and CHEN CHEN and PU WANG (2022) *EdgeML: Towards network-accelerated federated learning over wireless edge* (Computer Networks) [doi.org/10.1016/j.comnet.2022.109396](https://doi.org/10.1016/j.comnet.2022.109396)
- [11] MURSHED, M. G. SARWAR and MURPHY, CHRISTOPHER and HOU, DAQING and KHAN, NAZAR and ANANTHANARAYANAN, GANESH and HUSSAIN, FARAZ (2019) *Machine Learning at the Network Edge: A Survey* (ACM Computing Surveys), 54th vol
- [12] CHRIST, MAXIMILIAN and BRAUN, NILS and NEUFER, JULIUS and KEMPA-LIEHR, ANDREAS (2019) *Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh - A Python package)* (Neurocomputing) [10.1016/j.neucom.2018.03.067](https://doi.org/10.1016/j.neucom.2018.03.067)
- [13] NAKAMURA, TETSUYA and MAKIO, KOSUKE and UEHARA, KUNIYUKI (2006) *Discovering and Translating Skills from Motion Data*
- [14] NATH, MAHENDRA PRASAD and MOHANTY, SACHI NANDAN and PRIYADARSHINI, SUSHREE BIBHUPRADA B. (2021) *Application of Machine Learning in wireless Sensor Network* (2021 8th International Conference on Computing for Sustainable Global Development (INDIACom))
- [15] MOHANTY, SACHI NANDAN (2021) *Machine Learning for Healthcare Applications*
- [16] SEO, JEONG-WOO and KIM, TAEHO and LEE, JINSOO and KIM, JUNGIL and CHOI, JIN SEUNG and TACK, GYERAE (2019) *Fall prediction of the elderly with a logistic regression model based on instrumented timed up & go* (Journal of Mechanical Science and Technology), 33rd vol
- [17] ZEMBLYS, RAIMONDAS and NIEHORSTER, DIEDERICK and HOLMQVIST, KENNETH (2016) *Detection of oculomotor events using random forest*
- [18] SANCHEZ, VERALIA and SKEIE, NILS-OLAV (2019) *Decision Trees for Human Activity Recognition in Smart House Environments* [10.3384/ecp18153222](https://doi.org/10.3384/ecp18153222)
- [19] SHIVRAJ, PRAPULLA and SINHA, DIPAYAN and C, GURUPRASAD and SHOBHA, G. and T C, THANUJA (2019) *Multi Mobile Agent Based Remote Health Monitoring*