

Employing machine learning techniques for activity recognition in training with human body landmark data

Aplicação de técnicas de aprendizado de máquina para o reconhecimento de atividades com dados de pontos de referência do corpo humano

Rubens de Oliveira Santos

Universidade FUMEC
rubensgomes34@gmail.com

Eric de Paula Ferreira

Universidade do Estado de Minas Gerais
eric.p.f@gmail.com

Fernando Silva Parreiras

Universidade FUMEC
fernando.parreiras@fumec.br

Fábio Corrêa

Universidade FUMEC
fabiocontact@gmail.com

Vinícius Figueiredo de Faria

Universidade FUMEC
vffconsultoria@gmail.com

Renata de Souza França

Universidade do Estado de Minas Gerais
profrenatafranca@gmail.com

Dárlinton Barbosa Feres Carvalho

Universidade Federal de São João del-Rei
darlinton@acm.org

ABSTRACT

Machine learning methods for recognizing human activities have achieved remarkable results recently. Following this trend, this research aims to investigate automated activity recognition in the context of students undergoing training at UniverCemig. This quantitative-exploratory research applied the Support Vector Machine (SVM), K-Nearest Neighbors, Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) techniques. As a result, K-Nearest Neighbors and SVM achieved 91% and 96% performance, respectively, while GRU and LSTM achieved 96% performance for both. In conclusion, automated recognition of human activities is promising and suitable for training students at UniverCemig, helping instructors during training and, consequently, bringing agility to the student's learning certification process.

Keywords: Human activity recognition. Machine learning. Deep machine learning.

RESUMO

Métodos de aprendizado de máquina para o reconhecimento de atividades humanas alcançaram resultados notáveis recentemente. Seguindo essa tendência, esta pesquisa visa investigar o reconhecimento automático de atividades no contexto de alunos em treinamento na UniverCemig. Esta pesquisa quantitativa-exploratória aplicou as técnicas de Máquina de Vetores de Suporte, K-Nearest Neighbors, Unidade Recorrente com Portões (Gated Recurrent Unit) e Memória de Longo Curto Prazo (Long Short-Term Memory). Como resultado, K-Nearest Neighbors e Máquina de Vetores de Suporte alcançaram desempenhos de 91% e 96%, respectivamente, enquanto Unidade Recorrente com Portões e Memória de Longo Curto Prazo alcançaram 96% de desempenho para ambos. Em conclusão, o reconhecimento automático de atividades humanas é promissor e adequado para o treinamento de alunos na UniverCemig, ajudando instrutores durante o treinamento e, conseqüentemente, trazendo agilidade para o processo de certificação de aprendizado do aluno.

Palavras-chave: Reconhecimento de atividade humana. Aprendizado de máquina. Aprendizagem de máquina profundo.

1. INTRODUCTION

Human Activity Recognition (HAR) is an active research field in computer vision that aims to automatically detect information about activities performed by people through digital devices (POPPE, 2010). With technological advancements, various devices such as RGB cameras, depth cameras, RGB-D cameras, and Microsoft Kinect have been developed, enabling the extraction of different types of information, among which human body landmarks stand out (TASNIM et al., 2020).

The extraction of human body landmarks is a technique that has been applied within computer vision to perform HAR, for example, to estimate the poses of a human being (STRAKA et al., 2011), identify human behavior (SAPINSKI et al., 2019), and facial expression detection (FILNTISIS et al., 2019).

Machine Learning (ML) is a valuable technique for recognizing human activities and has driven innovation in many processes. For instance, ML can automatically recognize performed activities, assisting in a student training and certification process. The data of the performed activities are extracted after training. Adopting automated tools that perform tasks such as HAR contributes to a swift certification process.

For better performance, ML model creation relies on specific data, such as image processing of students undergoing training in activities of interest. Based on these reflections, this study aims to assess computational techniques to automate and recognize human activities with data obtained from images of activities performed by a human being under training – the electrician training field offered by UniverCemig. To achieve this objective, this research will be divided into three steps: a) create a dataset; b) train ML models for HAR classification; and c) compare the results of the implemented algorithms. Following this introduction, the text is

structured as follows: (section 2) theoretical review of the themes that support the study; (section 3) methodological procedures; (section 4) analysis of the results; and (section 5) final considerations.

2. THEORETICAL FRAMEWORK

2.1 HUMAN ACTIVITY RECOGNITION

Computer Vision is the scientific field focused on developing theories and methods for the automatic extraction of useful information contained in images captured by devices such as video cameras and scanners (CROWLEY; CHRISTENSEN, 1995). It is a subfield of artificial intelligence aimed at artificially modeling human vision by performing actions through its functions with the development of advanced software and hardware.

Human Activity Recognition (HAR) involves identifying or predicting human activities based on sensor data (DEHGHANI et al., 2019) or image capture (VAHORA; CHAUHAN, 2019), utilizing machine learning techniques for inference. Data acquired from a user's movements are captured and converted into characteristic patterns through a combination of hardware and software components. After data capture, the activity performed is inferred.

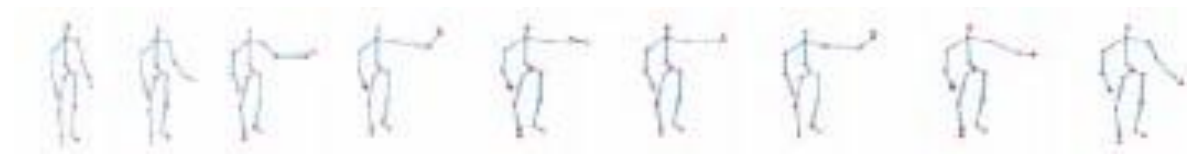
According to Lara and Labrador (2013), HAR is a temporal series problem. The data used refers to the movement performed by the user at a certain moment. Thus, HAR can be approached in the following ways: 1) sequential time series approach; and 2) sliding window time series approach.

The sequential time series approach can be derived from a data set $S = S_0 \dots S_{k-1}$, where k is defined by a time series and each feature is contained within a time interval $l = [t\alpha, t\omega]$. The goal is to look at the time partition $\langle I_0 \dots I_{r-1} \rangle$ of l , according to the data in the set S and find the classes that represent the performed activity. The data are consecutive, cannot be empty, and do not overlap. This definition implies that a person does not perform two activities at the same time (LARA; LABRADOR, 2013).

The sliding window time series approach can be given by the data set $W = W_0 \dots W_{m-1}$ for m being the size of the sliding window, fully or partially labeled, for each W_i containing a set of time series, given by $S = S_0 \dots S_{k-1}$ and a set $A = a_0 \dots a_{n-1}$ of labeled activities, where the goal is a mapping function $f: S_i \rightarrow A$ that can be evaluated for all possible values of S_i , so that $f(S_i)$ is as similar as possible to the actual activity performed during W_i (LARA; LABRADOR, 2013).

Human movement is coordinated by different parts of the body, and technology allows for representation through coordinates obtained from each point of the human body, which, in a dataset, represents a performed action, as shown in Figure 1. Such representation can be defined as a set of coordinates used to describe a person's pose in a specific temporal space (MANZI et al., 2017).

Figure 1. Human Body Landmarks



Source: research data.

Key points that make up a human body can be extracted by a depth camera with a specific tracker or using another model that maps the image. The skeletal model of the human body varies according to the software and device adopted.

A skeletal model can be modeled with several joint points (SHOTTON et al., 2011), where each point describes coordinates in relation to the position of each limb (MANZI et al., 2017). According to Tasnim et al. (2020), data models can be created from the joint points extracted from the skeletal model to form a dataset that can be used by machine learning techniques to perform HAR.

2.2 MACHINE LEARNING

Machine Learning (ML) is a subfield of artificial intelligence aimed at exploring data models and building algorithms that can autonomously understand data. Using ML algorithms, through a set of input data and based on certain patterns, it is possible to infer the outputs. According to Sugiyama and Kawanabe (2012), the main purpose of ML is to build systems that can adapt to constantly changing environments and learn from their experience. Another definition by Mitchell et al. (2013) reinforces the previously stated, characterizing ML as a computer system created to optimize performance using data and past experience. Machine learning can be divided into several categories, such as:

- **Supervised Learning:** in this category, ML receives all input data labeled as a training set and makes predictions about previously unseen data. A disadvantage of this category is that most of the time, data are manually labeled to guide the learning process (DAVILA et al., 2017).
- **Unsupervised Learning:** unlike the supervised method, this category does not know the output data (MITCHELL et al., 2013).
- **Semi-supervised Learning:** this category uses both labeled and unlabeled data. The advantage of this category is that only a limited number of labeled samples is needed (DAVILA et al., 2017).

Some machine learning methods are Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN). According to Rosati et al. (2018), SVM is a binary classifier capable of differentiating data between two classes. Its structure is based on statistical theory, aiming to propose machine learning techniques to maximize the model's generalization capacity and minimize structural risk. For classification, data are projected into a multidimensional space, enabling classification between two classes.

In the multidimensional space, the separation between classes is achieved with a hyperplane that maximizes its distance from the so-called support vectors. According to Janidarmian et al. (1995), SVM consists of sets of supervised learning models and associated algorithms that analyze data used for classification and regression problems.

According to Janidarmian et al. (1995), from a dataset, it is possible to construct a model capable of learning from new examples, assigning new classes to these examples, aiming to separate distinct categories by as wide a margin as possible to minimize the generalization error of the classifier. This model works with two perspectives, the classification of linearly separable and non-separable classes.

A dataset can be considered linearly separable when its data can be separated in a multidimensional plane. Therefore, when the model of Janidarmian et al. (1995) finds a hyperplane that perfectly separates the data from each class and whose separation margin is as large as possible, this hyperplane is called the optimal hyperplane. Margin is the region delimited by a pair of parallel hyperplanes that separate the two data classes with the greatest possible distance.

When the training data are linearly separable, it is possible to find a rigid margin hyperplane that meets the requirement of maximum separability as long as there are no errors and/or outliers in the data. However, if the data are not linearly separable, or if a certain tolerance for classification errors is desired (aiming to avoid overfitting), a soft margin hyperplane, which will not fully adjust and will potentially have a superior generalization capacity, should then be found. In this case, it is necessary to find a hyperplane that minimizes the probability of classification error.

According to Sáez et al. (2016), K-Nearest Neighbors (KNN) is a machine learning algorithm for classification and regression. The method is based on the premise that examples with similar characteristics should remain close. When a new example is added, KNN performs an analysis of the nearest neighbors, through the calculation of distance, to classify an unknown instance (AGUILETA et al., 2019). During classification, KNN analyzes the class with the lowest cost. If there is a class with the same lowest cost, then a random tie-break is performed among the groups with the lowest cost (JANIDARMIAN et al., 2017). The starting formula for KNN is: $eftKin = \sqrt[n]{n}$, where n is the number of training data samples.

The work published by McCulloch and Pitts (1943) initiated various studies on ANNs, also known as Deep Machine Learning (DML), as it sought to understand how neurons function. After this understanding, a computational model for neural networks based on mathematics and algorithms, called threshold logic, simulating the functioning of neurons, was created. Haykin (2001) classifies ANN as a powerful and complex machine capable of processing a large amount of information in minimal time and has the characteristic of simulating the brain's behavior when performing a task. For Janidarmian et al. (2017), the ANN is a system of interconnected neurons capable of creating mathematical models to perform machine learning.

According to Haykin (2001), ANN is formed by basic processing units, called perceptrons, understood as a decision-making unit with several inputs and a single output. McCulloch and Pitts (1943) classify a neuron as an information processing unit that is fundamental to the operation of the brain. With technological advances and the evolution of ANN concepts, various architectures for machine learning have been created, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

To solve the problem of gradient explosion or vanish gradient, Hochreiter and Schmidhuber (1997) proposed the LSTM architecture. This model is capable of learning and storing long-term dependencies by storing information in an internal memory (WANG et al., 2017). LSTM uses the concept of gates: a mechanism based on the multiplication of the input, which defines the behavior of each memory cell individually (ORDONEZ; ROGGEN, 2016).

Proposed by Chung et al. (2014), GRU is capable of storing data and weights in memory, having an architecture similar to LSTM; however, with a simpler implementation and computing power. Its architecture is formed by two gates, an update (z) and a reset (r), both passing information through many data windows, seeking to achieve better prediction or classification (LEI et al., 2018). GRU was also created to solve the problem of gradient explosion or vanish gradient.

3. METHODOLOGICAL PROCEDURES

This research is characterized as applied. According to Gil (2008), applied research aims to investigate, generate knowledge for practical application, prove or refute hypotheses. The nature of this research is quantitative. The quantitative approach aims to ensure accuracy of the analyzed data, thereby increasing the confidence margin on these for decision-making using statistical techniques (BAPTISTA; CUNHA, 2007). In this research, statistics will be used during the analysis and results phase.

The research has an exploratory character. As Gil (2008) states, exploratory research is characterized by developing, clarifying, and modifying concepts and ideas with a view to formulating problems or researchable hypotheses for subsequent studies. The exploratory research character is justified due to the nature of this research, which aims to implement machine learning techniques capable of recognizing activities performed by a human being.

The locus of this research is UniverCemig. This venue establishes a set of essential activities for electrician training. However, for this research, data were collected from an activity called B1, which involves assembling a structure. In this activity, the student must follow all the procedures and norms developed in the theoretical part of the course, and these data are collected for the application of machine learning techniques SVM and KNN, GRU, and LSTM.

Thus, this study seeks to verify, through metrics, the results of the implemented machine learning techniques, which involve comparisons of any nature requiring a definition of the perspective from which the data should be analyzed (CORDOVA; SILVEIRA, 2008). Human

activity recognition in the context of electrician training must correctly label activities since it could otherwise harm the student's learning certification process. In this manner, using accuracy, precision, and recall metrics to evaluate machine learning models is suitable, corresponding to the absence of false-positive and false-negative results.

4. RESULTS

The sections below detail the configurations of the techniques assessed and the achieved results, which have been divided into two sections: Experiments with Machine Learning (ML) techniques and Experiments with Artificial Neural Networks (ANN) techniques, also referred to as Deep Machine Learning (DML).

4.1 EXPERIMENTS WITH MACHINE LEARNING TECHNIQUES

This section presents the results and configurations from the implementations of two ML techniques, KNN and SVM. The techniques were trained with collected data, separated into test and training sets, 25% and 75%, respectively. The best hyperparameters were achieved through manual testing. A window of 44 samples per second was chosen, with time varying between 3 and 60 seconds of data for feature extraction that can capture cycles in activities. To reduce information loss at the window edges, a 50% overlap was used. According to Su et al. (2014), relevant data can be found through calculations that group them, returning information that can be utilized by ML techniques. Thus, for each axis, the square root, standard deviation, maximum, and minimum were calculated (SU et al., 2014; NGUYEN et al., 2015).

Various manual tests were conducted to select the best parameter values, with the value of k varying between 3 and 20. Accordingly, the KNN parameters for executing the KNN techniques are presented in Table 1.

Table 1. KNN Hyperparameters

Parameter	Value
$K_{\text{neighbors}}$	8
Weights	uniform
Metric	minkowski
P	1

Source: research data.

The best SVM results were achieved manually using four functions to test the kernel: radial, linear, polynomial, and sigmoid. For the C function, tests varied between 1 and 10. The SVM parameters that showed the best results are presented in Table 2.

Table 2. SVM Technique Configuration Hyperparameters

Parameter	Value
Kernel	polynomial
C	5
Gamma	1

Source: research data.

Table 3 displays the experiment's results for our model in terms of precision, recall, and accuracy for the UniverCemig dataset. High precisions of approximately 97% with SVM and 95% with KNN were achieved. Tests were conducted with time windows between 3 and 160 seconds.

Table 3. Results of SVM and KNN Techniques

Time Window (Seconds)	Algorithm	Accuracy	Recall	Precision
3(S)	KNN	0.9175	0.9176	0.9178
	SVM	0.9657	0.9657	0.9658
5(S)	KNN	0.9020	0.9026	0.9015
	SVM	0.9328	0.9328	0.9329
10(S)	KNN	0.8750	0.8750	0.8895
	SVM	0.9459	0.9454	0.9472
20(S)	KNN	0.8571	0.8671	0.8675
	SVM	0.8918	0.8917	0.8975
30(S)	KNN	0.8378	0.8378	0.8412
	SVM	0.9183	0.9183	0.9339
60(S)	KNN	0.8887	0.8888	0.9206
	SVM	0.9583	0.9583	0.9642

Source: research data.

The SVM technique showed better results compared to KNN, even as the time window increased. According to Tasnim et al. (2020), performance loss occurs because a larger time window can capture more information, which can significantly affect classifier performance. The SVM technique was not affected by the time window; however, KNN showed a slight performance drop when the window was between 10(S) and 60(s).

4.2 EXPERIMENTS WITH DEEP MACHINE LEARNING TECHNIQUES

This section will detail the configurations and results of two DML techniques for performing HAR. The dataset-generated files were separated into training and testing, with 3 files for training and 1 for testing, with a margin of 31% and 69% for testing and training, respectively. The experiment of the proposed techniques started defining the architectural style

of each one, after which the search for the best batch size and number of epochs that would present the best accuracy began. The experiment started with epoch number 1 and gradually increased until a better result was achieved. The architectural style proposed by each model, their results, how the training was conducted, and the settings will be detailed below.

After defining the architectural style, as presented in Table 4, the execution of the experiment will begin. During training, the time window values were alternated between 1 and 6 seconds, and the number of epochs was set at 500 with a batch size of 300. The patience parameter was also set at 30, meaning that if the model stops learning, the training execution will immediately stop after 30 attempts.

Table 4. GRU Model Hyperparameter Configuration

Layer	Activation Function	Number of Neurons	Dropout Percentage
GRU	Relu	32	
GRU	Relu	32	0.5
GRU	Relu	16	0.5
Dense	Sigmoid	1	

Source: research data.

Similar to GRU, the first step was to define the architectural style of LSTM, as seen in Table 5. After this, training began, varying the time window between 1 and 6 seconds, the number of neurons and layers to find the best training time and learning for the model. The training was conducted with 500 epochs and a batch size of 300. During the training, it was observed that adding a larger number of layers and neurons made the model learn faster; however, no improvement in results was noticed. This could be explained by the amount of data in the Univercemig dataset. Compared to the GRU training result, it was noticed that this model learned faster but used more computational power, increased training time, and did not show a difference in results for the dataset.

Table 5. LSTM Model Hyperparameter Configuration

Layer	Activation Function	Number of Neurons	Dropout Percentage
LSTM	Relu	16	
LSTM	Relu	16	0.2
LSTM	Relu	8	0.5
Dense	Relu	100	
Dense	Sigmoid	1	

Source: research data.

As observed in Table 6, accuracies of approximately 96% were achieved with LSTM and GRU when reaching a time window value of 6 seconds. When comparing the results presented by the models, no significant difference was noticed. However, LSTM utilized more computational resources in terms of layers and neurons, leading to faster learning but not reflected in better results.

Table 6. Results of GRU and LSTM Techniques

Time Window (Seconds)	Algorithm	Accuracy	Recall	Precision
6(S)	LSTM	0.9681	1	1
	GRU	0.9681	1	1
5(S)	LSTM	0.8051	1	1
	GRU	0.8053	1	1
4(S)	LSTM	0.6971	1	1
	GRU	0.6972	1	1
3(S)	LSTM	0.7725	1	1
	GRU	0.7725	1	1
2(S)	LSTM	0.5493	0.50	0.4140
	GRU	0.5493	0.3532	0.3532
1(S)	LSTM	0.3514	0.3334	0.1171
	GRU	0.3513	0.3332	0.1171

Source: research data.

The GRU architecture used fewer computational resources and a smaller number of layers and neurons; however, its result was not altered. Both applied models supported a larger amount of data, as expected; however, the data window to achieve good results was 360 seconds. This could also be explained by the amount of data in the UniverCemig dataset. For both models, activation functions were varied in the input and output layers. The Relu activation function showed better performance in the input layers, and the Sigmoid function also presented the best performance in the output layer.

5. FINAL CONSIDERATIONS

The objective of this research was to implement ML techniques to recognize activities performed by UniverCemig students during their training process. This study highlights the potential for implementing activity recognition in educational settings, offering substantial support to instructors, and facilitating the student certification process.

The findings suggest that deep learning techniques excel when analyzing larger temporal windows — window size exceeded 6 seconds — while SVM and KNN showed significant efficacy across various window sizes. All are suitable for the UniverCemig dataset. SVM, notably, provided exceptional outcomes across all temporal variations, positioning it as an advantageous option for data analysis.

This research was limited to addressing the recognition of human training activities performed by UniverCemig students based on the B1 structure assembly activity. However, for future work, it is necessary to collect more data and apply the techniques to achieve good performance and reduce the time window.

REFERENCES

- Aguileta, A., Brena, R., Mayora, O., Molino Minero Re, E., and Trejo, L. 2019. "Multisensor fusion for activity recognition - a survey," *Sensors* (19:17), pp. 1-14.
- Baptista, S. and Cunha, M. 2007. "Estudo de usuários: visão global dos métodos de coleta de dados," *Perspectivas em Ciência da Informação* (12), pp. 168-184.
- Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. 2014. "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proceedings of the NIPS 2014 Deep Learning and Representation Learning Workshop*, pp. 1-14.
- Gil, A. C. 2008. "*Métodos e técnicas de pesquisa social*," 6 ed., São Paulo: Editora Atlas.
- Crowley, J. I., Christensen, H. I. 1995. "Vision as process: basic research on computer vision systems," J. I. Crowley, H. I. Christensen (eds.); with editorial assistance by: Alain Chehikian e. a., pp. 341–371. Berlin: Springer, 1995.
- Davila, J., Cretu, A.-M., and Zaremba, M. 2017. "Wearable sensor data classification for human activity recognition based on an iterative learning framework," *Sensors* (17:6), pp. 1287.
- Dehghani, A., Sarbishei, O., Glatard, T., and Shihab, E. 2019. "A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors," *Sensors* (19:22), pp. 5026.
- Filntisis, P., Efthymiou, N., Koutras, P., Potamianos, G., and Maragos, P. 2019. "Fusing body posture with facial expressions for joint recognition of affect in child–robot interaction," *IEEE Robotics and Automation Letters* (4:4), pp. 4011–4018.
- Haykin, S. 2001. "*Redes Neurais: Princípios e prática*," São Paulo: Bookman Editora.
- Hochreiter, S. and Schmidhuber, J. 1997. "Long short-term memory," *Neural computation* (9), pp. 1735– 80.
- Janidarmian, M., Roshan Fekr, A., Radecka, K., and Zilic, Z. 1995. "Support-vector networks," *Machine learning* (20), pp. 273-297.
- Lara, O. D. and Labrador, M. A. 2013. "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys Tutorials* (15:3), pp. 1192-1209.
- Lei, T., Zhang, Y., Wang, S., Dai, H., and Artzi, Y. 2018. "Simple recurrent units for highly parallelizable recurrence," Preprint, pp. 1-15.
- Manzi, A., Dario, P., and Cavallo, F. 2017. "A human activity recognition system based on dynamic clustering of skeleton data," *Sensors* (17:5), pp. 1-14.
- McCulloch, W. S. and Pitts, W. 1943. "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*. Cambridge, Massachusetts, USA, 1 edition.

- Mitchell, R., Michalski, J., and Carbonell, T. 2013. “*An artificial intelligence approach*,” Cambridge, Massachussets, USA, 1 Edition.
- Nguyen, P., Akiyama, T., Ohashi, H., Nakahara, G., Yamasaki, K., and Hikaru, S. 2015. “User-friendly activity recognition using svm classifier and informative features,” *IEEE*, pp. 1-8.
- Ordonez, F. and Roggen, D. 2016. “Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition,” *Sensors* (16:1), pp. 115.
- Poppe, R. 2010. “A survey on vision-based human action recognition,” *Image and Vision Computing* 28:6), pp. 976-990.
- Rosati, S., Balestra, G., and Knaflitz, M. 2018. “Comparison of different sets of features for human activity recognition by wearable sensors,” *Sensors* (18:12), pp. 4189.
- Sapiński, T., Kamińska, D., Pelikant, A., and Anbarjafari, G. 2019. “Emotion recognition from skeletal movements,” *Entropy* (21:7), pp. 646.
- Sáez, Y., Gómez, A. B., and Viñuela, P. I. 2016. “A comparison study of classifier algorithms for cross-person physical activity recognition,” *Sensors* (17:1), pp. 66.
- Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. 2011. “Real-time human pose recognition in parts from single depth images”, In *Proceedings of the CVPR 2011*, pp. 1297–1304.
- Straka, M., Hauswiesner, S., Rütther, M., and Bischof, H. 2011. “Skeletal graph based human pose estimation in real-time,” In *Proceedings of the BMVC 2011*, p. 1-12.
- Su, X., Tong, H., and Ji, P. (2014). “Activity recognition with smartphone sensors,” *Tsinghua Science and Technology* (19:2), pp. 35-249.
- Sugiyama, M. and Kawanabe, M. 2012. “*Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*,” volume 4 of 10. The name of the publisher, Cambridge,MA, USA, 1 edition.
- Tasnim, N., Islam, M. M., and Baek, J.-H. 2020. “Deep learning-based action recognition using 3d skeleton joints information,” *Inventions*, (5:3), pp. 49.
- Vahora, S. and Chauhan, N. 2019. “Deep neural network model for group activity recognition using contextual relationship, *Engineering Science and Technology, an International Journal* (22:1), pp. 47- 54.
- Wang, Y., Velswamy, K., and Huang, B. (2017). “A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems,” *Processes* (5:3), pp. 1-15.