Anomaly Detection in Smart Houses: Monitoring Elderly Daily Behavior For Fall Detecting

Yves M. Galvão, Vinícius A. Albuquerque, Bruno J. T. Fernandes, Mêuser J. S. Valença
Polytechnic School of Pernambuco - POLI, University of Pernambuco - UPE, Pernambuco - Brazil
Email: (ymg,vaa,bjtf,mjsv)@ecomp.poli.br

***Abstract*—Smart Houses and Internet of Things (IoT) are two present tendencies in our days. Due to these technologies, the existent types of equipment in a smart house (sensors, thermostats, and video cams) allow us to analyze and collect data from a person's daily activities and use it in the field of anomaly detection. Therefore, noninvasive monitoring techniques can be applied to people's residences. When focusing on the elderly population, this type of approach can be used to detect and report a fall, decreasing the costs of monitoring these individuals. This paper uses images from a Microsoft Kinect cam, accelerometers' data, digital image processing and computer vision techniques to make a comparative study between different supervised classifiers and statistic approaches when they are being used in the fall detection problem. The results show that some of the tested classifiers are efficient in this task, reaching an accuracy of 96.67% and 98.79%.**

***Keywords*—anomaly detection; iot; internet of things; knn; mlp; svm; moving average; fall detection; smart houses;**

I. Introduction

The concept of residential automation exist since 1960 [1], but in that decade the devices were insufficient, only executing simple tasks. The birth of new technologies and smart devices, caused this old concept called smart house to reborn with strength in this last decade [2].

Internet of Things (IoT) is another concept that helped on the developed of smart houses. One of the reasons for this was the release of new devices and applications by the big companies like Apple, Samsung, and Amazon [3], allowing the use of sensors in residences.

IoT can easily be applied to smart houses [4], allowing users to remotely control all the devices directly connected to this house [5].

A relevant application to IoT in smart houses is a noninvasive people monitoring system [6]. This type of application is advantageous when it comes to houses of individuals with some disability [7], or any person who needs constant attention, such as a percentage of the older adults.

The extracted sensors' data used in the IoT concept allow us to analyze the typical behavior of people on daily basis activities and then use anomaly detection techniques to detect unexpected situations [8], [9]. Using this method, it is possible to use devices already present in a smart house, reducing the high costs of implementation of monitoring systems [8] and making the monitoring of a person in an automatic and noninvasive way.

978-1-5386-3734-0/17/$31.00

In the elderly monitoring, the fall detection can be vital in a smart house. A fall represents a great health risk for elderly [10]. A large of techniques provide support to detect falls, one example is the wearable approach, using the accelerometer sensor data [11]–[13]. In a smart house, the sensors and cameras help to detect anomalies [14]–[16].

After the evaluation of the experiments, the results of classifiers and statistical approach achieve a high rate of accuracy, reaching an accuracy of 96.67% and 98.74%.

This paper is organized into four main sections. Section II details the used methodology for the experiments. The section III shows the parameters used and the results achieved in the tests. Finally, section IV presents the conclusions and future work.

II. Methodology

This section is focused on showing the methods used for the anomaly detection approach used in this paper. The model of the proposed technique can be seen in the Figure 1.

A. Acquisition

A Microsoft Kinect camera and accelerometers are used for the data acquisition. The images captured by the camera consist of 30 falls and 40 activities of daily living (ADL) and the accelerometer data contain 30 falls.

B. Preprocessing

The images preprocessing of the dataset was performed to allow the feature extraction and to generate the data for the supervised tests. The following steps are were executed:

- Image transformation to grayscale (256).
- Histogram equalization
- For the time series: Every three frames are calculated the Mean Squared Error (MSE) described in II-C1.
- For the supervised tests: Image resizing to 40 x 40 pixels.

C. Feature extraction

The feature extraction is defined by the calculation of the MSE it is applied on the images after the preprocessing step.

1) Mean Squared Error: MSE corresponds to a fidelity measure. The objective of calculating this metric is to measure the fidelity of one image concerning another image. The result is a quantitative score of fidelity/similarity [17].

The data are obtained by the calculation of two different images and follow the following equation: MSE 1.

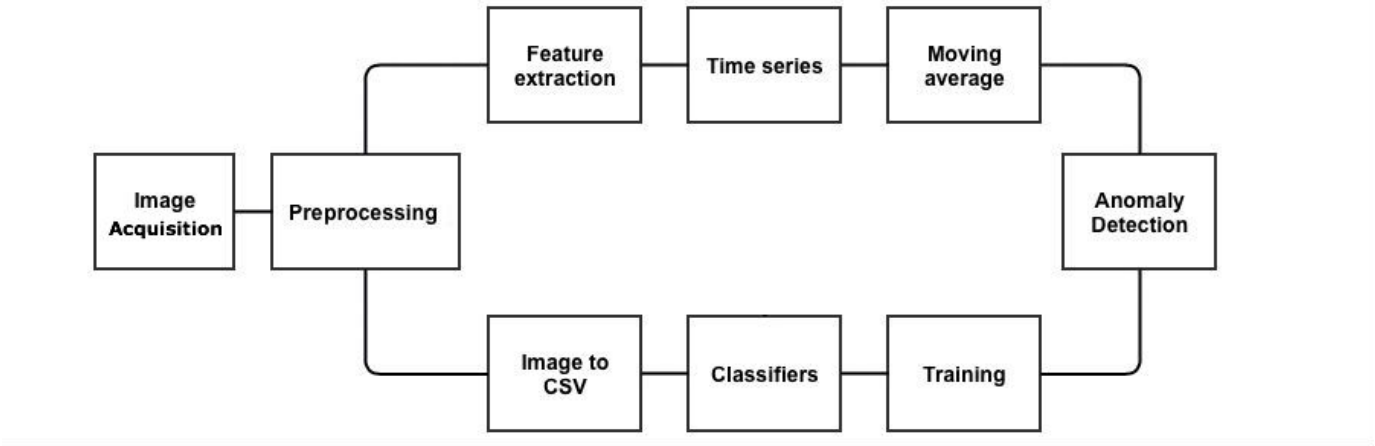

Fig. 1. Anomaly Detection flow.

$$MSE[img1, img2] = \frac{\sum((img1 - img2)^2)}{length(img1)} \quad (1)$$

2) Accelerometers: The extracted data of the accelerometers are used in comparative with the extracted data of the captured images (MSE).

All captured data are defined in unity gravity (g) for the time stamp of the event. The equation 2 represents the calculation of the final value of one isolated event.

$$SV_{total} = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (2)$$

D. Time series

The cited techniques in II-C1 and II-C2 allow the feature extraction and the creation of a time series that, in conjunction with the moving average equation, have identified anomaly points in a determined window.

E. Moving average

The calculation function of moving average is utilized together with a linear discrete function, according to the equation 3.

$$f \times g(t) = \int_{-\infty}^{\infty} f(T) \times g(t - T)dT \quad (3)$$

One point is considered an anomaly if the calculated value is below X_1 or greater than X_2. To calculate X_1 and X_2, the following equation is used 4.

$$X_1 = \mu - (\sigma \times \phi) \, | X_2 = \mu + (\sigma \times \phi) \quad (4)$$

F. Classifiers

The following classifiers were used in the labeled tests: Multi Layer Perceptron (MLP), K-Nearest Neighbors (KNN) and Support Vector Machines (SVM). The used parameters in each classifier are described in section III.

1) Multi Layer Perceptron - MLP: It is a group of multiple layers that has two distinct states and process its connected elements using sigmoid [18]. Between the input layer and output layer, there are neurons in one or more hidden layers, these layers have influence in the final classification, tuning the output weights. Valença [19] describes an MLP network as a generalization of one perceptron network, with the addition of one or more hidden layers. The Figure2 shows an example of MLP.

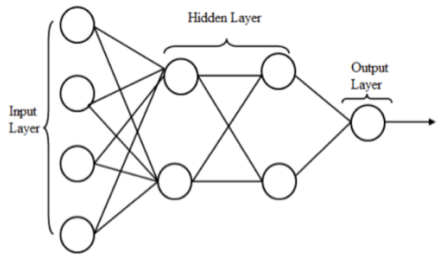

Fig. 2. An example of MLP network.

2) K-Nearest Neighbors - KNN: Deekshatulu et al. [20] define KNN as a simple algorithm which stores all cases and classify new cases based on a similarity measure. The most simple algorithm of KNN is described in two steps: 1) find the K training instances which are closest to an unknown instance. 2) pick the most commonly occurring classification for these K instances [20]. For its simplicity, KNN is widely used in the classification problems. The Figure 3 shows an example of KNN classifier.

3) Support Vector Machine - SVM: This technique was developed based on the theory of statistic learning [21]. The first algorithm was proposed by Cortes and Vapnik in 1995 [22].

The main objective of an SVM is to determine decision borders in the available space, isolating the data points on different classes. So the classifier creates a plane in the center of one calculated margin with the objective to minimize the

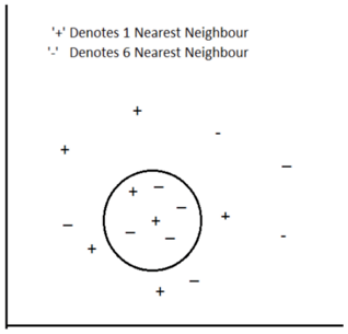

Fig. 3. Result of one KNN classifier.

error generalization [21]. An SVM classifier can be viewed in the Figure 4.

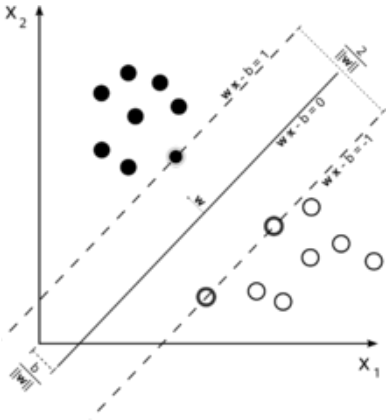

Fig. 4. An SVM classifier result.

III. Experiments

In this section, it is shown the dataset, classifiers and the results obtained in this paper.

A. Dataset

The following dataset was employed in the experiments for anomaly detection: UR Fall Detection Dataset [23]. We can split the dataset into three different parts:

- Dataset 1: contains the captured image frames with the ADL and fall registers.
- Dataset 2: contains the data of the accelerometers.
- Dataset 3: contains the extracted features from the image frames.

The dataset 1 contains a sequence of captured frames by one video cam. In the statistics approach experiments, the video camera 0 of the dataset was utilized. This dataset contains 30 falls register in different situations. The dataset 2 contain the accelerometers data synchronized frame by frame with the dataset 1.

The calculations of MSE is obtained after the preprocessing images of the dataset 1, described in the II-C1.

The number of used frames is defined after different tests with values between 1 and 15. For this experiment, the value 3 was the most efficient. The figure 5 shows an example of a dataset image.

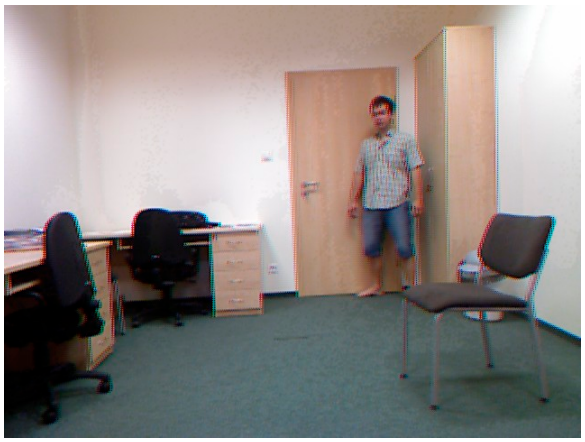

Fig. 5. Example of the dataset image.

In the Figure 6 it is possible to verify the original image and the image after preprocessing, respectively.

Fig. 6. Frame before and after preprocessing

For the classifiers tests, the dataset 3 is utilized. This dataset contains a total of 2995 instances, split into 2074 normal classes and 921 anomalies. The process of feature extraction can be verified in II-C1.

Every image pixel is used as input attribute of the classifiers. At the end of every line, one label is added to indicate the class of the image, being 0 to normal and 1 to anomaly.

B. Metrics

This section describes the approach that was utilized for the comparison of the used techniques.

1) Statistic approach:

- The data are plotted in a time series, and the moving average algorithm is executed for the anomaly points detection.
- The identified points are compared with the correspondent frame of captured image.
- When a point is wrongly detected, the whole sample is considered to be wrong.
- At the end of the experiment, the total score of success is calculated according to the equation 5 and the result is compared with the dataset two results.

$$\%ACCURACY = \frac{E \times 100}{N} \quad (5)$$

$Where$
$E:\ Quantify\ of\ identify\ erros\ in\ N$
$N:\ total\ of\ examples$

2) Supervised tests:

- Preparation of the dataset, adding the following labels, 0 for normal and 1 to anomaly.
- Classifier training using different combinations of parameters until a good percentage of success is obtained.
- At the end of the experiment, the total score of success is calculated according to the equation 5 and the result is compared with the results of dataset 2. In the supervised tests, the F1SCORE is also calculated according to the equation 6.
- The results obtained are compared between the different classifiers (KNN, MLP e SVM).

$$F_1 = 2 \times \frac{P \times r}{P + r} \quad (6)$$

$Where$
$P:\ It\ is\ defined\ as\ the\ number\ of\ true\ positives(T_p)\ over\ the\ number\ of\ true\ positives\ plus\ the\ number\ of\ false\ positives(F_p).$
$R:\ It\ is\ defined\ as\ the\ number\ of\ true\ positives(T_p)\ over\ the\ number\ of\ true\ positives\ plus\ the\ number\ of\ false\ negatives(F_n).$

C. Experimental settings

Different tests were realized with different parameters for the classifiers, as it was previously said. The next section describes in detail the used parameters.

- MLP
  The learning rate was changed between 0.1 and 0.5. In the next step, the number of hidden layers was modified, varying from 1 to 5. Finally, different activation functions were tested (Sigmoid, hyperbolic tan and linear function).
  - Activation Function: Sigmoid
  - Learning Rate: 0.5
  - Hidden Layers: 1
  - Epochs: 500
  - hidden neurons: 4
- SVM
  Only eps and type of the kernel were changed in the tests. The kernels utilized were: RBF, linear, polynomial and sigmoid.
  - Kernel: Polynomial
  - Cost: 1
  - Eps: 0.001
  - Coef: 0.5
- KNN
  In the KNN classifier, the only parameter that changed in the tests was the number of K. The values tested were the values between 1 and 15. The best results were obtained with the following combination of parameters:
  - K: 2
  - Cross Validate: True
  - Search Algorithm: Linear Search

All tests were executed with K-fold cross-validation where the number of folds is 3.

For the tests with moving average and accelerometers, σ values between 1 and 10 and window size between 5 and 30 were tested. In the tests with moving average utilizing MSE, the variation of σ was between 1 and 10 and window size between 10 and 200 for the tests. The best combination of parameters is described in the Table I.

TABLE I
PARAMETERS USED WITH MOVING AVERAGE

Dataset	Parameters
Accelerometers	window size = 20 $\sigma = 4$
MSE	windows size = 90 $\sigma = 2.8$

D. Results

The results for the classifiers MLP, KNN and SVM are in the Table II and the confusion matrix for every experiment are described in Tables III to IV.

TABLE II
RESULTS FOR CLASSIFIERS

Classifier	Accuracy	F1SCORE	Cost for classification
KNN	98.79%	0.991	0.24s
MLP	98.79%	0.988	0.01s
SVM	98.39%	0.988	0.01s

TABLE III
CONFUSION MATRIX - KNN

		CLASSIFIED	
		FALL	NO_FALL
ACTUAL	FALL	889	32
	NO_FALL	4	2070

TABLE IV
CONFUSION MATRIX - MLP

		CLASSIFIED	
		FALL	NO_FALL
ACTUAL	FALL	902	19
	NO_FALL	17	2057

The MLP obtained the highest accuracy and the lowest number of False Negatives, so, evaluating by this parameters,

TABLE V
CONFUSION MATRIX - SVM

		CLASSIFIED	
		FALL	NO_FALL
ACTUAL	FALL	901	20
	NO_FALL	28	2046

it would be considered the best between the tested classifiers for this task. Furthermore, the classification time for the SVM and MLP classifiers were approximately 24x faster than the KNN, although the F1Score is higher in the KNN classifier the number of False Negatives for fall is lower in MLP.

The results using moving average can be seen in the figures 7 and 8 respectively. In the figures, it is possible to identify anomaly points above the graphic, identified by a star.

In the 30 dataset examples, both acquisition techniques had a rate superior to 96% for anomaly detection. The use of accelerometers had a detection rate of 96,67% of the fall cases, while the use of MSE identified one incorrect point at the video 9, getting the same score. The results can be verified in the table VI.

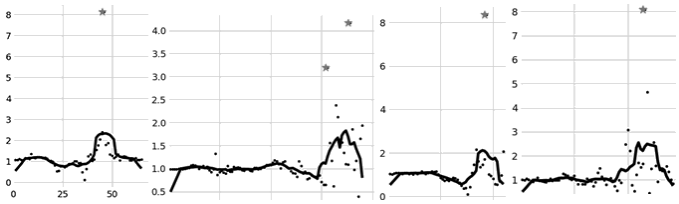

Fig. 7. Result using moving average with the accelerometers data

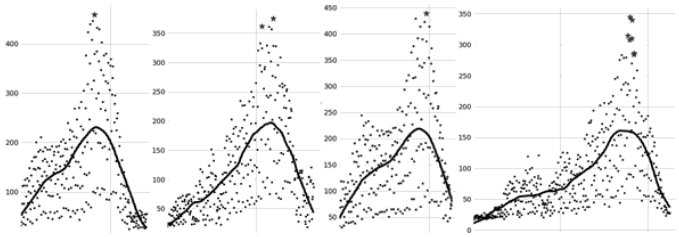

Fig. 8. Result using moving average with the MSE data

TABLE VI
RESULTS FOR MOVING AVERAGE

Dataset	Accuracy
Accelerometers	96,67%
MSE	96,67%

Although the use of accelerometers data has the same accuracy of the MSE, this method needs a physical device in the body of the resident, being a disadvantage in comparison with the utilization only a Kinect camera and feature extraction (MSE). The device can be forgotten to be worn by the subject or it can be a discomfort in his/her day.

In comparison with the supervised techniques, the moving average presents a bit advantage, since it does not need a training phase.

IV. CONCLUSION

This paper presented possible solutions for the anomaly detection using moving average and supervised classifiers to detect falls. The techniques were applied to the UR Fall Detection dataset with different classifiers: MLP, KNN, and SVM for labeled tests and moving average in time series.

The results showed that the utilized techniques reached an accuracy score above 96% in fall detection. Even though the proposed model achieved this level of accuracy, in this paper, we had the limitation of not having the data for falls for older adults, which are the main target for this type of research.

For future works:

- To use more recent classifiers methods, such as deep learning.
- Use of proposed model in a real smart house, for the test efficiency of the developed model.

REFERENCES

[1] R. Harper, *Inside the smart home*. Springer Science & Business Media, 2006.

[2] M. Miller, *The Internet of things: How smart TVs, smart cars, smart homes, and smart cities are changing the world*. Pearson Education, 2015.

[3] J. Hong, J. Shin, and D. Lee, "Strategic management of next-generation connected life: Focusing on smart key and car–home connectivity," vol. 103. Elsevier, 2016, pp. 11–20.

[4] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "Iot gateway: Bridgingwireless sensor networks into internet of things," pp. 347–352, 2010.

[5] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C.-H. Lung, "Smart home: Integrating internet of things with web services and cloud computing," vol. 2, pp. 317–320, 2013.

[6] A. Agarwal, S. Dawson, D. McKee, P. Eugster, M. Tancreti, and V. Sundaram, "Detecting abnormalities in iot program executions through control-flow-based features," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. ACM, 2017, pp. 339–340.

[7] M. Chetouani, J. Cohn, and A. A. Salah, "Hbu: International workshop on human behavior understanding," in *Human Behavior Understanding*. Springer, 2016, pp. 339–340.

[8] V. Jakkula, D. J. Cook *et al.*, "Anomaly detection using temporal data mining in a smart home environment," vol. 47, no. 1. Stuttgart [etc.] FK Schattauer [etc.], 2008, pp. 70–75.

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," vol. 41, no. 3. ACM, 2009, p. 15.

[10] A. F. Ambrose, G. Paul, and J. M. Hausdorff, "Risk factors for falls among older adults: a review of the literature," vol. 75, no. 1. Elsevier, 2013, pp. 51–61.

[11] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[12] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4544–4553, 2015.

[13] J. Howcroft, J. Kofman, and E. D. Lemaire, "Review of fall risk assessment in geriatric populations using inertial sensors," *Journal of neuroengineering and rehabilitation*, vol. 10, no. 1, p. 91, 2013.

[14] U. Bakar, H. Ghayvat, S. Hasanm, and S. Mukhopadhyay, "Activity and anomaly detection in smart home: A survey," in *Next Generation Sensors and Systems*. Springer, 2016, pp. 191–220.

[15] E. Hoque, R. F. Dickerson, S. M. Preum, M. Hanson, A. Barth, and J. A. Stankovic, "Holmes: A comprehensive anomaly detection system for daily in-home activities," pp. 40–51, 2015.

[16] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and V. Bulgari, "From lab to life: Fine-grained behavior monitoring in the elderly's home," pp. 342–347, 2015.
[17] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," vol. 26, no. 1. IEEE, 2009, pp. 98–117.
[18] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on neural networks*, vol. 3, no. 5, pp. 683–697, 1992.
[19] M. J. Valença, "Fundamentos das redes neurais: exemplos em java," 2016.
[20] B. Deekshatulu, P. Chandra *et al.*, "Classification of heart disease using k-nearest neighbor and genetic algorithm," *Procedia Technology*, vol. 10, pp. 85–94, 2013.
[21] P. C. Deka *et al.*, "Support vector machine applications in the field of hydrology: a review," vol. 19. Elsevier, 2014, pp. 372–386.
[22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
[23] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," vol. 117, no. 3. Elsevier, 2014, pp. 489–501.