IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 9, 2021, accepted May 16, 2021, date of publication May 24, 2021, date of current version June 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3083064

A Framework for Anomaly Identification Applied on Fall Detection

YVES M. GALVÃO[1], LETÍCIA PORTELA[1], JANDERSON FERREIRA[1], PABLO BARROS[1,2], ROBERTA ANDRADE DE ARAÚJO FAGUNDES[1], AND BRUNO J. T. FERNANDES[1], (Member, IEEE)

[1]Escola Politécnica de Pernambuco (POLI), Universidade de Pernambuco (UPE), Recife 50720-001, Brazil
[2]Cognitive Architecture for Collaborative Technologies (CONTACT) Unit, Istituto Italiano di Tecnologia, 16163 Genova, Italy

Corresponding author: Yves M. Galvão (ymg@ecomp.poli.br)

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brasil under Grant 001, and in part by the Brazilian agencies are Fundação de Amparo a Ciência e Tecnologia do Estado de Pernambuco (FACEPE) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ).

ABSTRACT Automatic systems to monitor people and subsequently improve people's lives have been emerging in the last few years, and currently, they are capable of identifying many activities of daily living (ADLs). An important field of research in this context is the monitoring of health risks and the identification of falls. It is estimated that every year, one in three persons older than 65 years will fall, and fall events are associated with high mortality rates among the elderly. We propose an anomaly identification framework to detect falls, which incorporates a spatial-temporal convolutional graph network (ST-GCN) as a feature extractor and uses an encoder process to reconstruct ADLs and identify falls as anomalies. As the publicly available fall datasets are few and generally unbalanced, training a reliable model using approaches that need explicit labeling is challenging. Thus, a focus on learning without external supervision is desirable. Treating a fall as an exception of ADLs allows us to recognize falls as anomalies without explicit labels. Given its modular architecture, our framework can robustly represent visual information and use the encoder's reconstruction error to identify falls as anomalies. We assess our framework's ability to recognize falls by training it with only ADLs. We perform three types of experiments: single dataset training and evaluation that consists of separate 90% of the data to train the model 5% to adjust the model, and the rest to the test. A joint dataset experiment, where we combine two datasets to increase the number of samples our model is trained on, and a cross-dataset evaluation, where we train on one dataset and evaluate using another one. Besides presenting state-of-the-art results on our experiments, particularly on the cross-dataset one, the model also presents a low number of false events, which makes it an ideal candidate for real-world application.

INDEX TERMS Anomaly detection, autoencoders, deep learning, encoders, fall detection, ST-GCN.

I. INTRODUCTION

It is estimated that every year, one in three people older than 65 years will fall [1], and, if no improvements are made, the number of falls is estimated to increase 100% by 2030 [2]. Falls are most commonly occur inside a home environment [3], and thus, the recognition of activities of daily living (ADLs) is an essential related field. Correctly identifying a fall is a difficult task since there are many similar ADLs, e.g., activities related to lying on the floor, lying on the bed, or even some workout exercises [4]. Furthermore, there are only a few public datasets related to health risks because it is an expensive task to collect this data and might be a strong ethical violation of personal and sensitive information [5]. In addition to all of that, not only is the fall itself a danger to people's health, but the damages of falling can be amplified for people who live alone: as a person can remain on the floor for an extended period of time, resulting in several health problems, e.g., dehydration, internal bleeding, or even death [1]. Thus, efficient automatic accident reporting is essential to mitigate these effects.

An automatic system to monitor people and quickly report falls can reduce the extension of the injuries caused by the fall and reduce the chances of long-term damage and death [1]. Training a model that can distinguish falls from ADLs can be challenging because the available fall datasets are usually

The associate editor coordinating the review of this manuscript and approving it for publication was Tony Thomas.

unbalanced, having several ADLs but few fall events [6]–[8]. In particular, for approaches that need explicit labeling, unbalanced datasets lead to biased models with a high number of false alerts. Also, in an indoor environment, some sensory input variations such as lighting variance and occlusions can hinder the correct identification a fall [9].

The most common approaches to fall identification are based on the acquisition and processing of accelerometer information from wearable-sensors, such as wristbands [10], [11] watches [12], or even smartphones [13], [14]. In this context, generally, simple classifiers such as K-Nearest Neighbour (K-NN) [15], or a Support Vector Machine (SVM) [16] are enough to identify falls. However, in some cases, they usually present many false-negative events [15]. A solution for this problem was the proposal of a bidirectional-Long Short-Term Memory (bi-LSTM) with a soft fusion which reduces the number of false alerts [17]. Still, in real-world environments, models based on active sensors fail to provide reliability because people can forget to wear the device, specially the elderly or people who have dementia [18].

Cameras are a better solution to using wearable sensors use, allowing for the monitoring of a home environment without relying on active sensors. Although, compared with wearable sensors, this approach generally has a high initial cost for monitoring one individual. In some environments, solutions based on the use of cameras are more appropriate and do not depend on people using an active sensor. In this context, a Convolutional Neural Network (CNN) is a common approach for activity recognition on videos [19], [20]. To improve the robustness of automatic fall detection systems based on videos [21], a variety of different CNNs topologies have been proposed. For example, a multimodal network combining CNN and LSTM to process the temporal video data [20], or another based on RGB-D, which uses the background extraction in a pre-process phase [22]. The latter tries to avoid the sensory input variation, focusing on the people, but uses an approach that needs explicit labeling in an unbalanced dataset. Because of that, it fails to improve real-world applications' reliability.

Categorizing falls as anomalies is a possible solution to circumvent the problem of unbalanced datasets. Thus, a model that does not need a label to learn has an advantage, as it is not biased by the label's distribution, identifying anomalies without the need for previous event knowledge. In this case, one straightforward approach is the use of One-Class classifiers [23], [24]. The main objective of this approach is to train a model with the ADL instances alone. Consequently, when a fall instance is presented to the model, it can identify the event as an anomaly. However, in a high-dimensional dataset, the model requires a large amount of memory and high computational power to be trained [25]. The use of autoencoders is also common in unbalanced datasets [26]–[28]. Similar to the One-Class models, autoencoders can be trained using only ADL instances. They can reconstruct these instances with a minimal reconstruction error, identifying anomalies when the reconstruction error is greater than a defined threshold. However, because of the high-dimensionality of images, pure autoencoders suffer from learning robust data representations [29].

The use of autoencoders in fall detection problems is not a novelty [30]–[32]. However, develop a model that presents a generalization between differents datasets is a challenge. In [31] the authors suggest a Deep Spatio-Temporal Convolutional Autoencoders using three different public datasets [6], [33], [34]. Although the results suggest an accuracy superior to the 97% in the SDU dataset, the model suffers to reach a good accuracy in the UR-Fall dataset, which has a small number of samples. Another example is the Cai *et al.* model [32], which provides a convolutional autoencoder using a novel method based on the hourglass convolutional auto-encoder (HCAE-FD). The paper shows state-of-art results in comparison with similar works using the same dataset. Still, it fails to provide reliability because the model was only tested in one dataset, which has a small number of samples.

Using an already trained model as a feature extractor provides a robust feature representation of ADL actions, allowing the creation of robust fall detection systems [35], [36]. To avoid fine-tuning the model in different environments, a solution capable of learning using only ADLs is more suitable than labeled solutions. Nowadays, the use of CNNs as a specialized feature extractor is employed in different works [37]–[39]. Yhdego *et al.* [35] proposed a pre-trained AlexNet to create an automatic fall detection system. They retrained the last three layers of the AlexNet using a small dataset [6]. Using this approach, the authors' proposed model presented a better result than other existing models based on SVMs in the same dataset. However, as this model represents a frame using a frame-level static posture, it loses contextual information from a sequence of actions [40]. An alternative to avoid these limitations is using temporal features, i.e., ST-GCN, which can provide strong feature representation based on ADLs [41]. The use of the body's skeleton information for fall detection problems was tested in different works [42]–[45]. The ST-GCN model uses RGB or RGB-D videos as input, along with a skeleton estimation, and it can generate a robust spatial-temporal representation.

In this paper, we propose a framework to recognize falls as anomalies from an ADL perspective. Once trained, this framework can identify falls as anomalies when the reconstruction error of the stimulus's representation is greater than a pre-defined threshold. We used the ST-GCN as a feature extractor in the framework's pre-processing phase to obtain a robust and efficient representation from a sequence of RGB images. In the evaluation phase, an encoder is trained to reconstruct ADLs and consequently identify anomalies based on the reconstruction error.

We evaluate our framework using three different datasets: UP-Fall [8], UR-Fall [6], and PRECIS HAR [7]. The datasets have different representations of ADLs and falls performed in an indoor environment; some also present light variance and occlusions, especially the UR-Fall dataset. We performed three types of experiments: single dataset training and

evaluation to verify the model's learning capability. A joint dataset experiment, where we combine two datasets to increase the number of samples to verify if the high-dimensional dataset enhances the model's learning capability, and a cross-dataset evaluation to verify the model's generalization. Because of the datasets' imbalanced nature, a high accuracy not necessarily represent a good result, as the framework can classify all ADLs correctly and all falls actions incorrectly, and still present a high accuracy. Thus, we decide to use the geometric mean, specificity, and sensitivity as the primary metrics.

Analyzing the results, we verify that our framework based on reconstructing errors presents a low number of false negative and false positive events in the single dataset experiments, reaching state-of-the-art in all of the evaluated datasets. Furthermore, our framework also performed well on cross-validation experiments, making it a strong candidate for automatically detecting fall systems in real-world applications.

This paper is presented as follows: Section II describes our proposed framework in detail. Section III details the experimental setup, i.e., the used datasets, the experiments, and the hyper-parameter tunning to define the best encoder architecture. In Section IV, we presented the obtained results and compared them with the current state-of-the-art. Lastly, Section V presents the conclusion and future works.

II. OUR PROPOSED FRAMEWORK

Our framework's primary purpose is to identify falls without the need for previous knowledge of these events. In this case, we have chosen to use an encoder model capable of identifying anomalies based on reconstruction error. As the public datasets have only a few samples of falls, we decided to use a pre-trained ST-GCN in a high-dimensional dataset as a feature extractor. The complete flow of our proposed framework can be seen in Fig. 1.

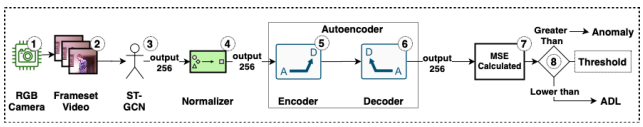


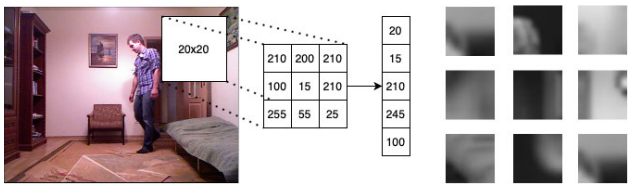
FIGURE 1. (1-2) We start capturing a frameset using an RGB camera; (3). The ST-GCN extracts features; (4). Normalizing the ST-GCN extracted data; (5). Encoding the extracted features; (6). Decoding the extracted features; (7). Calculating the Mean Squared Error (MSE) (8). If the reconstruction error is lower than the defined threshold, the event is an ADL; and if the reconstructed error is greater than the threshold, the event is a fall.

The first step of our framework is to capture a frameset of a video using an RGB camera. Then the entire frameset is used as input for the ST-GCN model. We normalize the data of the middle layer of the ST-GCN model. This step completes our framework's pre-processing phase, described in more detail in Section II-A. In the next phase, our framework tries to encode/decode the resulting data. So, we calculate the Mean-Squared Error (MSE), and the value obtained is compared with a threshold. If the calculated value is greater than the pre-defined threshold, an anomaly is identified; this process is described in detail in Section II-C.

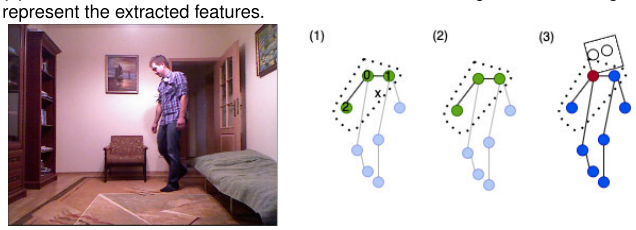
A. PRE-PROCESSING

In the pre-processing phase, we use ST-GCN as a feature extractor. The ST-GCN was trained using the Kinetics 400 dataset [46], which consists of an ADL dataset with 400 different classes captured by an RGB camera.

Unlike the CNN model, a GCN model can analyze the image and extract vectors, organizing them in a connected group of graphs. Fig. 2 show an example of feature extraction in a CNN network (2a) and an ST-GCN (2b).



(a) The CNN is used to extract features from an ADL image. The last images represent the extracted features.



(b) First, the ST-GCN model labels the nodes according to the distance from the gravity center, defined by × in the image (1). Then, nodes are connected according to their proximity (2). Finally, all nodes are connected, and the entire skeleton information is available (3).

FIGURE 2. The difference between the features extracted by CNN and GCN.

The spatial convolutional graph is defined in (1).

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{tj}) \cdot w(l_{ti}(v_{tj})) \tag{1}$$

wherein Z is related to the subset's cardinality, and w is similar to a 2D convolution kernel, and represents the weight function. In the mapping function, l_{ti} maps to a node in its subset neighborhood label. The v_{ti} and v_{tj} represent the node of the graph, and the $B(v_{ti})$ the sampling area.

To improve the extracted data of the ST-GCN which present a high variance, we decided to normalize the data of all datasets using (2).

$$\begin{aligned} X_{norm^i} &= (X^i - X_{min^i}) * \frac{255}{(X_{max^i} - X_{min^i})}, \\ X_{calculated^i} &= \frac{(X_{norm^i} - 127.5)}{127.5}, \\ Z_i &= \frac{X_{calculated^i} - \overline{X}^i}{s^i}. \end{aligned} \tag{2}$$

where $\overline{X}^i$ is the mean of X_{norm^i}, X_{max^i} and X_{min^i} are respectively the maximum and minimum value of the current dataset. s^i represents the standard deviation of the sample, Z_i the normalized value, and X^i represents the current data.

B. ENCODER

To identify anomalies, we use an encoder model trained only with ADLs actions. The model tries to reconstruct the stimulus' representation of the pre-processing phase. Combining the original information and the output of the encoder, we calculated the reconstruction error using the Mean Squared Error (MSE) as follows: $\mathrm{MSE}[\mathrm{O_{data}}, \mathrm{DE_{data}}] = \frac{\sum((\mathrm{O_{data}} - \mathrm{DE_{data}})^2)}{\mathrm{length}(\mathrm{O_{data}})}$. O is the normalized data in step 4 of the flow, and DE is the decoded data referring to step 6. The MSE value is compared with a pre-defined threshold, and if the value is greater than that, an anomaly is identified. The threshold was automatically chosen using the mean of MSE in the validation set multiplied by a constant or, in some experiments, the max MSE value in the training phase.

C. EVALUATION

We chose the following parameters to evaluate our experiments: specificity, F1-score, precision, sensitivity, AUC (Area Under the ROC Curve), and accuracy. As a high accuracy in the anomaly detection problem does not necessarily represent a good result, the accuracy was used only in comparison to the current state-of-the-art results. In contrast, for internal experiments, a special attention was made to the geometric mean. The geometric mean is based on the following parameters: sensitivity and specificity, and is defined as follow: $(\prod_{i=1}^{n} x_i)^{\frac{1}{n}} = \sqrt[n]{x_1.x_2.x_n}$. Where, the value of $\sqrt[n]{x_1.x_2}$ represents the sensitivity and specificity values, and n is the total number of parameters. Lastly, to guarantee the results are statistically relevant, we performed each experiment ten times, and the values presented in this paper are the averages of the experiments.

III. EXPERIMENTAL SETUP

This section describes our adopted experimental setup, i.e., used datasets, types of experiments, the chosen parameters, and the adopted architecture.

A. DATASETS

We have selected three different datasets to validate our framework: UP-Fall [8], UR-Fall [6], and PRECIS HAR [7]. The selected datasets contain several types of ADLs and falls. In the next sections, each selected dataset is briefly described.

1) PRECIS HAR

PRECIS HAR is a human action recognition dataset with RGB videos and depth captures recorded on a 3D camera. It has 16 classes, each with 50 clips, one per subject.

Although each video contains only one person, some include small occlusions in the RGB stream and different lighting conditions. All actions are outlined in Table 1.

2) UP-FALL

This human action recognition dataset is specifically targeted towards fall detection. It uses cameras and sensors, and consists of 11 actions performed three times by 17 subjects, totaling 561 videos. As outlined in Table 2, 6 of the actions are ADLs and the remaining five falls.

TABLE 1. Types of Actions Presented on PRECIS HAR Dataset.

ADL	Anomaly
Stand up	Fall from bed
Sit down	
Sit still	
Read	
Write	
Cheer up	
Walk	Faint
Throw paper	
Drink from a bottle	
Drink from a mug	
Move hands in front of the body	
Raise one leg up	

TABLE 2. Types of Actions Presented on UP-Fall Dataset.

ADL	Anomaly
Walking	Falling forward using hands
Standing	Falling forward using knees
Sitting	Falling backwards
Picking up an object	Falling sideward
Jumping	Falling sitting in empty chair
Laying	

3) UR-FALL

UR-Fall is a fall detection dataset with 30 falls and 40 ADLs, totaling 70 videos. It is an RGB-D dataset recorded with cameras and accelerometers. Falls were recorded with two Microsoft Kinect cameras and one accelerometer, while ADLs were recorded with only one camera and accelerometer. Many actions are recorded with light variance, and many lie actions as outlined in Table 3. For our work, only the RGB layers were used.

TABLE 3. Types of actions presented on UR-Fall dataset.

ADL	Anomaly
Walking	Falling forward
Walking up stairs	Getting Up of a Chair
Walking down stairs	Falling backwards
Picking up an object	When Seated
Sitting down	When Standing
Praying	

B. SPLIT DATABASE FOR EXPERIMENTS

The framework was evaluated in three different ways to validate the experiments: single dataset validation, cross-dataset validation, and joint dataset validation. The purpose of this setup is to improve the framework's overall validation and avoid overfitting. As our work focuses on anomaly detection, all split validations contain only ADL information in the training and validation sets.

1) SINGLE DATASET VALIDATION

The first partition method is the percent split validation, one of the most common evaluation approaches, wherein the available data is split into three. The training set is used to adjust the model's weights, the validation set provides information used to adjust the model, and the test set evaluates the final model.

Our proposed anomaly detection approach uses 90% of the ADL data to train and none of the fall data. The validation set is used to adjust the threshold and comprises 5% of ADL, whereas again using no fall data. Finally, the test set uses the remaining ADL data and all fall data to assess the model using the selected threshold.

2) CROSS-DATASET VALIDATION

This type of split uses training data from one dataset and tests with a different dataset. Because those datasets contain distinct types of images, it provides further information on the model's ability to generalize. It illustrates if the model is capable of applying knowledge abstracted from one database to another. The dataset division is made using 95% of one dataset to train the model and 5% to adjust it. Then, all falls are used in the test phase, combined with all of the other dataset's data.

3) JOINT DATASET VALIDATION

This validation technique combines datasets and uses the resulting information to both train and test the model. The resulting model has the most varied and extensive training and testing data, thus providing further evaluation opportunities. The dataset division is performed equally in both datasets. Thus, 90% of each dataset's ADL information is used to train the model, 5% is used to adjust the model, and another 5% combined with all fall events is used to evaluate the model.

C. HYPER-PARAMETER TUNNING

To define the architecture of the framework's autoencoder, we tried different numbers of layers, dense units, activation functions, batch size, and number of epochs, using Hyperopt [47]. For each type of experiment, we ran Hyperopt with two hundred evals, ranging the parameters according to the Fig. 3.

For each eval, Hyperopt trained the encoder with a random parameterization, using the dataset split according to the experiment type. Then, to define the experiment's threshold value, the mean MSE value of the validation set is calculated for the autoencoder model or the max MSE value for SVD and PCA. The MSE value is multiplied by a constant C. The constant C value is also tuned by Hyperopt, which tests values between 1.0 and 2.0. In the final phase, the model evaluates the data using the test set, and the AUC score is calculated. After all the trials, it returns the model that presented the highest AUC value.

We can see a heat map of each hyper-parameter tunning for the autoencoder model in Fig. 4. A black dot is plotted for each AUC obtained by the trial, and the blue star represents the best obtained AUC in all each trial. For the single

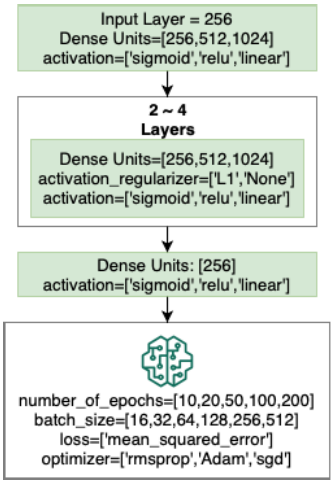


FIGURE 3. Attributes used for hyper-parameter tuning in the autoencoder model on the proposed framework.

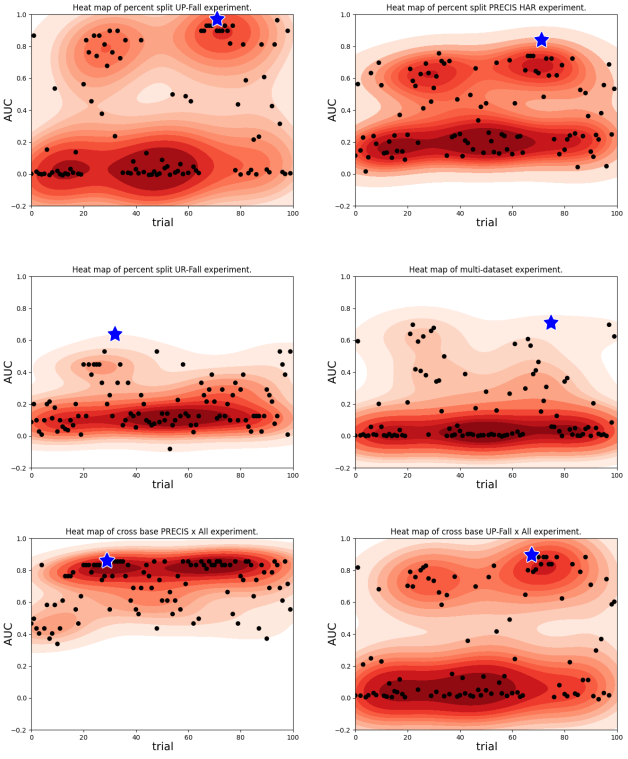


FIGURE 4. All Hyperopt trials for autoencoders. The blue star represents the best AUC.

dataset and joint experiments, we can see that in most trials, the presented AUC value is small (dark red regions), so it is concluded that the constant value C impacts the accuracy of the model directly. For the cross-dataset experiments, many trials presented AUC values close to the best-obtained AUC value (blue star) for different values of C.

We tried to apply the KPCA (Kernel PCA) to compare it to the results' autoencoder models. However, after trying different gamma values (varying between 0.01 and 5) and different kernels (linear, RBF, poly), all of the used datasets presented the best result using a linear kernel. Thus, we decided to compare the autoencoder's results to PCA and SVD.

For SVD and PCA encoders, we tried different numbers of components. A heat map of each hyper-parameter tuning can be seen in Fig. 5. and 6, respectively. Once again, the black dots represent the AUC value obtained in each trial, and the blue star the best AUC value obtained. For SVD and PCA encoders, Hyperopt presented the best results when using the max value of MSE on training data, and because of that, this was chosen for these encoders. Unlike the autoencoder model, the SVD model has presented less variance for the AUC value on each trial in all the experiments, except for the single dataset experiment in the UR-Fall dataset. Finally, for the PCA encoders, the impact of the constant C was only high in the single dataset experiment using the UR-Fall and in the joint dataset experiment.

As we can see, a key element in anomaly detection problems is the threshold value. Thus, we decided to plot a heat-map with small changes in the constant value C for each dataset in the single dataset experiment to compare it to our best value obtained. The results can be seen in Fig. 7.

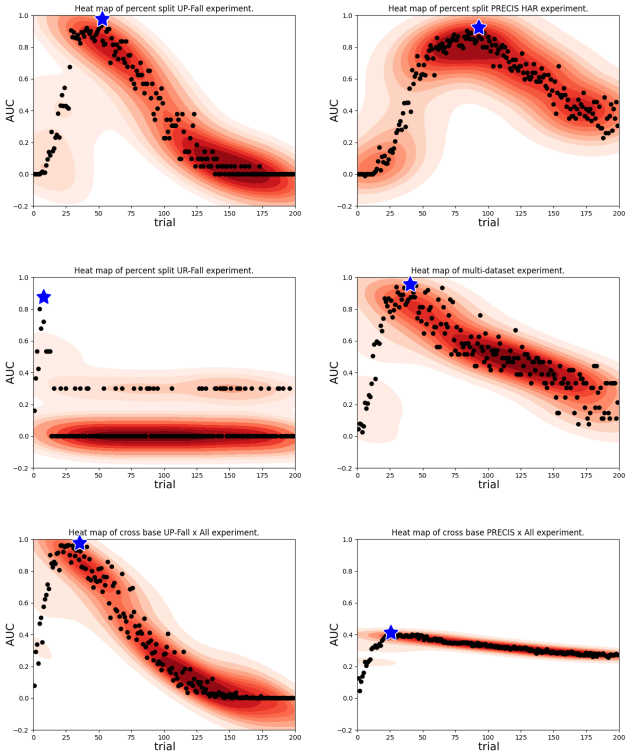


FIGURE 5. All Hyperopt trials for SVD. The blue star represents the best AUC.

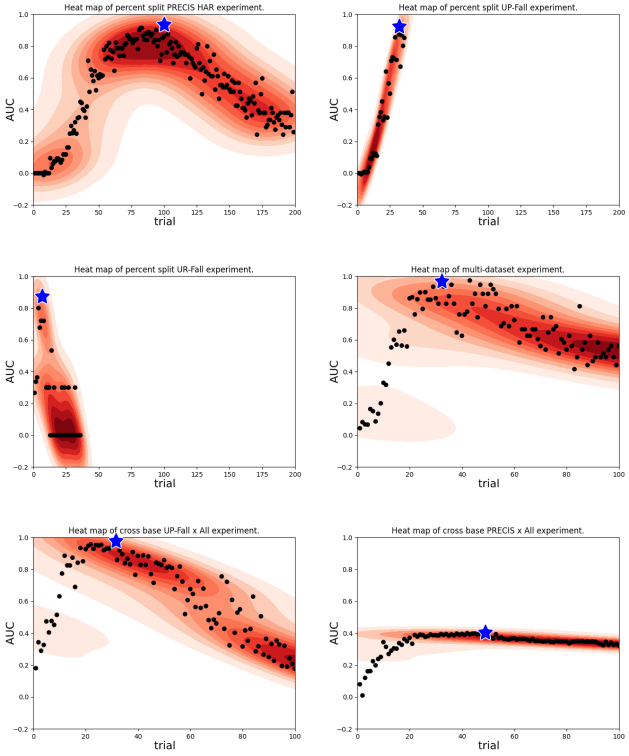
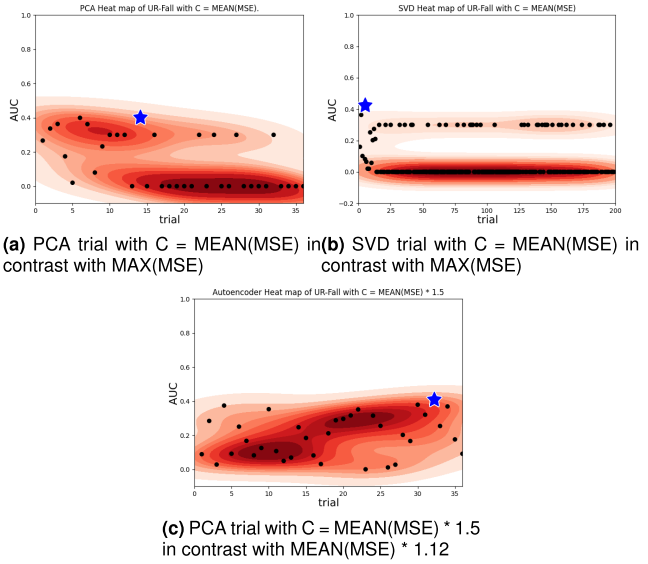


FIGURE 6. All Hyperopt trials for PCA. The blue star represents the best AUC.



(a) PCA trial with C = MEAN(MSE) in contrast with MAX(MSE)

(b) SVD trial with C = MEAN(MSE) in contrast with MAX(MSE)

(c) PCA trial with C = MEAN(MSE) * 1.5 in contrast with MEAN(MSE) * 1.12

FIGURE 7. One Hyperopt trial for SVD, PCA, and Autoencoder with different values for the constant C.

Finally, all the calculated Constant C values for the thresholds in each experiment type and dataset are outlined in Table 4.

TABLE 4. All calculated MSE and threshold constants for each experiment and dataset.

Encoder	Experiment	Threshold
Autoencoder	Percent Split - PRECIS HAR	$mean(MSE_{validation}) * 1.57$
	Percent Split - UR Fall	$mean(MSE_{validation}) * 1.12$
	Percent Split - UP Fall	$mean(MSE_{validation}) * 1.12$
	Joint	$mean(MSE_{validation}) * 1.3$
	Cross - PRECIS HAR vs. All	$mean(MSE_{validation}) * 1.3$
	Cross - UP Fall vs. All	$mean(MSE_{validation}) * 1.57$
SVD / PCA	Percent Split - PRECIS HAR	$max(MSE_{train})$
	Percent Split - UR Fall	$max(MSE_{train})$
	Percent Split - UP Fall	$max(MSE_{train})$
	Joint	$max(MSE_{train})$
	Cross - PRECIS HAR vs. All	$max(MSE_{train})$
	Cross - UP Fall vs. All	$max(MSE_{train})$

D. ARCHITECTURE OF OUR PROPOSED ENCODERS

We have analyzed different topologies and architectures using Hyperopt. The best architecture of autoencoders on PRECIS HAR, UP-Fall, UR-Fall, joint datasets, cross-dataset UP-Fall vs. PRECIS HAR, and cross-dataset PRECIS HAR vs. UP-Fall are in Fig. 8, and the SVD/PCA are in Fig. 9.

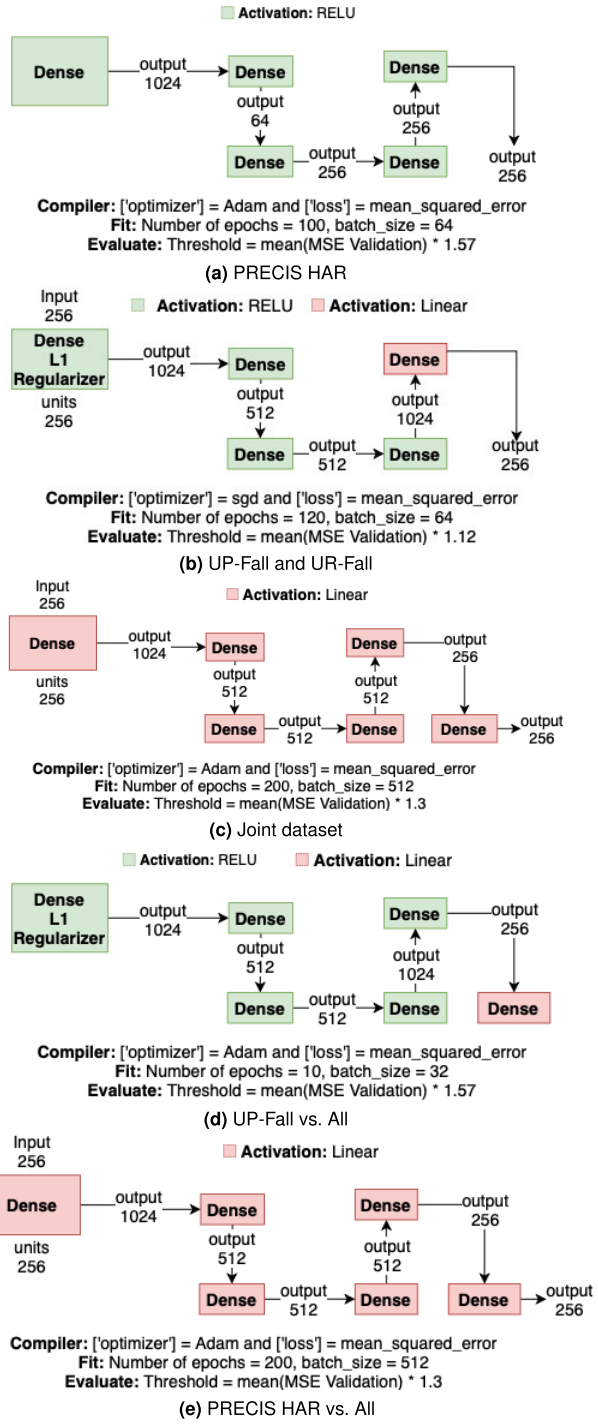


FIGURE 8. The final architecture of our proposed autoencoder in each experiment.

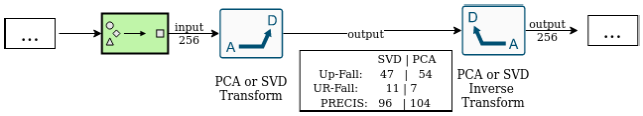


FIGURE 9. The final architecture of our proposed SVD/PCA model in each experiment.

IV. EXPERIMENTAL RESULTS

This section presents the obtained results after evaluated each type of split explained in Section III. Also, we have consolidated our best results and compared them to the current state-of-the-art. In Fig. 10, is described the summary of all our experiments. The following subsections summarize the results of each encoder.

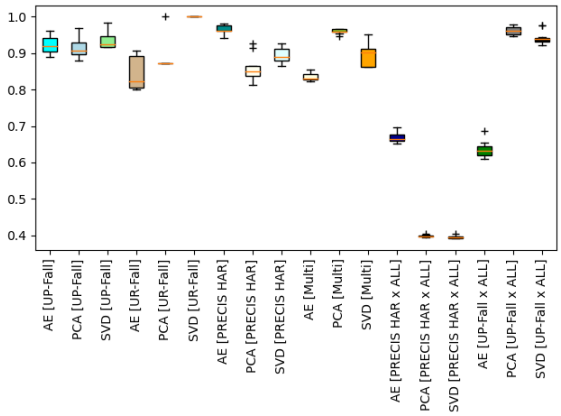


FIGURE 10. Boxplot of all obtained results. We can see less variance in the experiments that use PRECIS HAR dataset in the training phase. In contrast, experiments that use UP-Fall in the training phase present high variance.

TABLE 5. All obtained results for our proposed framework using the autoencoder model.

	PRECIS HAR	UP-Fall	UR-Fall	Joint	PRECIS HAR vs. ALL	UP-Fall vs. ALL
Accuracy	98.59%	98.52%	90.00%	96.68%	82.09%	81.09%
Geometric Mean	0.99	0.96	0.92	0.94	0.84	0.83
AUC	0.98	0.96	0.94	0.90	0.78	0.82
Specificity	0.99	0.99	0.98	0.98	0.81	0.82
Precision	0.96	0.92	0.90	0.82	0.78	0.81
F1	0.97	0.93	0.87	0.85	0.82	0.81
Sensitivity	0.99	0.94	0.86	0.90	0.88	0.83

A. AUTOENCODER

The autoencoder model's results are outlined in Table 5. Fig. 11, show the data's dispersion for all evaluated experiments.

Our framework reaches a high geometric mean value (greater than 0.9) for all three evaluated datasets in the single dataset experiment. Furthermore, the PRECIS HAR dataset and UP-Fall experiments present a small number of false-positive/negative events. The UR-Fall dataset also presents a high specificity of 0.98 and precision of 0.90.

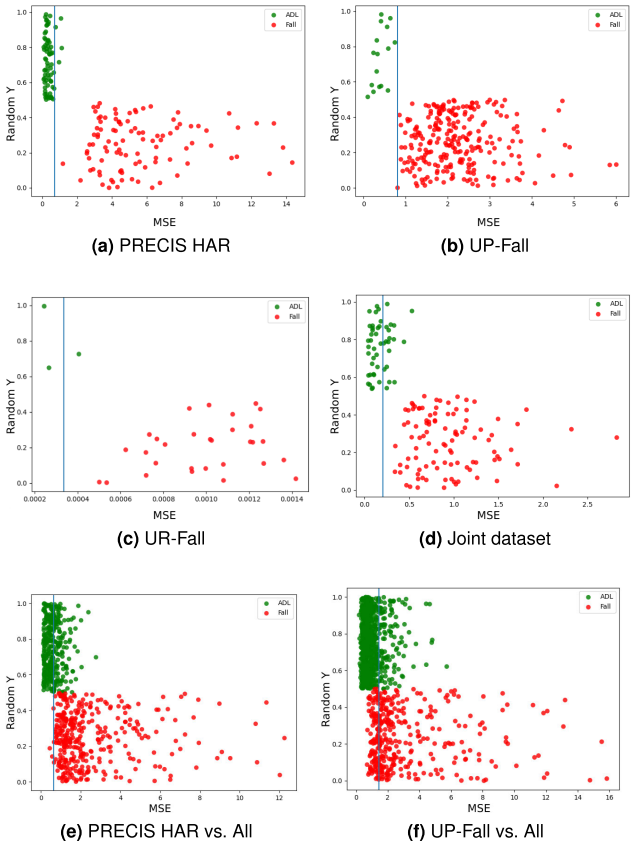


FIGURE 11. All reconstruction errors of ADLs and falls in all experiments, referring to the one trial. The blue line represents the threshold, and all plotted points to the line's right were classified as a falls and to it is left as ADLs.

The framework also reached an excellent geometric mean value (greater than 0.8) in the joint dataset experiment. The number of false events is greater than the single dataset results, however the numbers are not too high according to the specificity and sensitivity values.

Finally, on the cross-dataset validation experiments, the results are worse than the single dataset and the joint dataset experiments. However, they reached a greater geometric mean value on the PRECIS HAR vs. All and a substantial value (greater than 0.6) on the UP-Fall vs. All.

B. SVD

The SVD model results are outlined in Table 6, and Fig. 12 shows the dispersion of data for all evaluated experiments.

The SVD encoders present an excellent geometric mean value (superior to 0.9) in all experiments except in one the cross experiment: UP-Fall vs. All. Furthermore, in the UR-Fall percent split experiment, the SVD encoder reached the accuracy of 100%, identifying all ADLs and falls correctly.

C. PCA

The PCA model results are outlined in Table 7, and Fig. 13 also show the dispersion of data for all evaluated experiments.

TABLE 6. All obtained results for our proposed framework using the SVD model.

	PRECIS HAR	UR-Fall	UP-Fall	Joint	PRECIS HAR vs. ALL	UP-Fall vs. ALL
Accuracy	94.94%	100%	98.62%	96.62%	69.20%	97.39%
Geometric Mean	0.95	1.0	0.98	0.96	0.77	0.95
AUC	0.94	1.0	0.95	0.92	0.70	0.99
Specificity	0.93	1.0	0.99	0.96	0.61	0.92
Precision	0.89	1.0	0.90	0.85	0.41	0.96
F1	0.93	1.0	0.94	0.92	0.58	0.98
Sensitivity	0.98	1.0	0.98	0.96	0.99	0.99

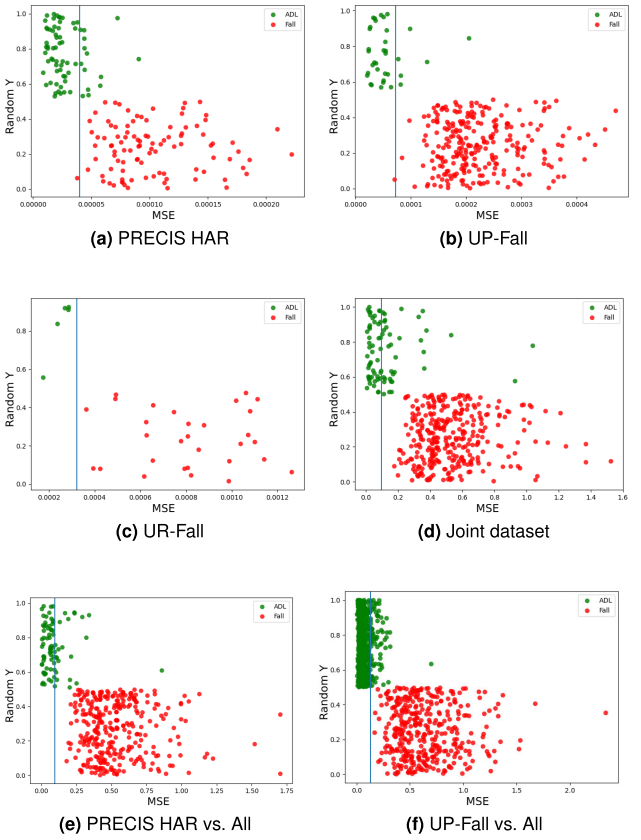


FIGURE 12. All reconstruction errors of ADLs and falls in all experiments, referring to the one trial. The blue line represents the threshold, and all plotted points to the line's right were classified as a falls and to it is left as ADLs.

Like the SVD encoder, the experiments using PCA also present excellent geometric mean values (superior to 0.9) in all experiments except in the cross dataset experiment UP-Fall vs. All.

D. COMPARISON BETWEEN OUR FRAMEWORK AND THE CURRENT STATE-OF-THE-ART

Looking for the current state-of-the-art for all used datasets, we verified that our proposed framework had beat the PRECIS HAR's accuracy, increasing it by 3.05%. In the UP-Fall dataset, although one work has a better accuracy

TABLE 7. All obtained results for the our proposed framework using PCA model.

	PRECIS HAR	UR-Fall	UP-Fall	Joint	PRECIS HAR vs. ALL	UP-Fall vs. ALL
Accuracy	98.59%	97.00%	98.52%	96.68%	69.30%	98.27%
Geometric Mean	0.99	0.98	0.96	0.94	0.78	0.97
AUC	0.92	0.91	0.93	0.98	0.70	0.99
Specificity	0.99	0.97	0.99	0.98	0.61	0.95
Precision	0.96	0.82	0.92	0.82	0.41	0.97
F1	0.97	0.90	0.93	0.85	0.58	0.98
Sensitivity	0.99	1.0	0.94	0.90	0.99	0.99

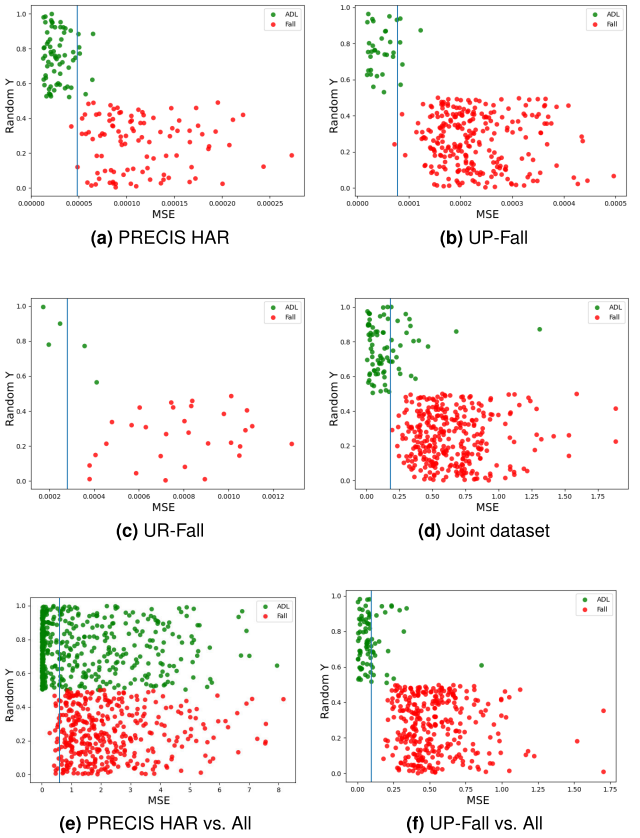


FIGURE 13. All reconstruction errors of ADLs and falls in all experiments, referring to the one trial. The blue line represents the threshold, and all plotted points to the line's right were classified as a falls and to it is left as ADLs.

than obtained with our model, it uses information from two different types of sensors in a multimodal approach [40], and because of that, it is outside of this work's scope. Our framework increased the accuracy by 2.69% compared to the UP-Fall dataset's state-of-the-art result in the single modality model, and it still achieved a geometric mean superior to 0.9. Finally, for the UR-Fall dataset, we increased the current state-of-the-art by 3.34%. Furthermore, our proposed framework was trained using only ADL events, allowing a better generalization of the problem and making it more suitable to real-world applications. In Table 8, we show all accuracy obtained with our framework and the current state-of-the-art

TABLE 8. Compilation of our Best Results Compared With the Current State-of-the-art with Single Sensor Input for Each Used Dataset.

UR-Fall				
Model	Accuracy	Sensibility	F1	Specificity
Our framework	**100%**	**1.0**	**1.0**	**1.0**
Model of Harrou *et al.* [48]	96.66%	1.0	0.96	0.94
Total of improvement:	**3.34%**			
UP-Fall				
Model	Accuracy	Sensibility	F1	Specificity
Our framework	**98.62%**	**0.92**	**0.93**	**0.99**
Model of Martínez-Villaseñor *et al.* [8]	95.93%	0.74	0.66	0.69
Total of improvement:	**2.69%**			
PRECIS HAR				
Our framework	**98.59%**	**0.95**	**0.97**	**0.97**
Model of Popescu *et al.* [49]	94,38%	-	-	-
Total of improvement:	**4.21%**			

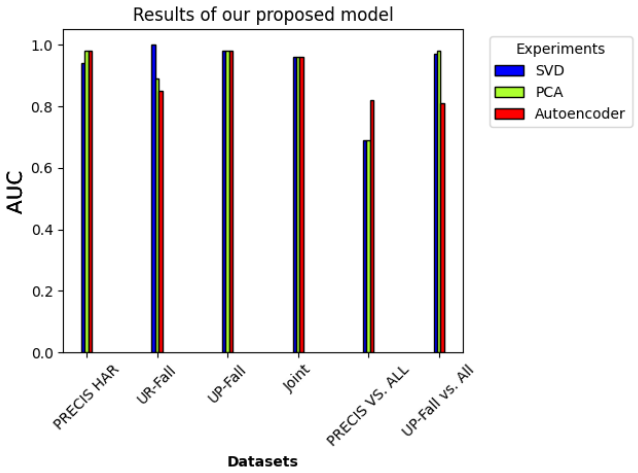


FIGURE 14. All obtained results for each encoder model in each experiment type.

presented in the literature, and in Fig. 14 we compile all of the obtained AUC values for each type of experiment and dataset.

E. DISCUSSION

The decision to evaluate our proposed framework in different types of experiments allowed us to verify the model's learning capacities and its generalization. Our framework presented high AUC values and excellent geometric mean values (greater than 0.8) in all single dataset experiments, reaching the state-of-art on the PRECIS HAR dataset, UR-Fall, and UP-Fall dataset. Furthermore, different from the current state-of-the-art, in the three evaluated datasets, our models have the advantage of only requiring ADL events in the training phase.

In addition to the experiments performed using the single dataset, we performed two additional experiments combining all datasets and cross-validation, i.e., training with only one dataset's ADL and evaluating with the other's. The results also presented excellent values for the geometric mean (greater than 0.8) in all experiments. It confirmed our initial suppositions about the generalization in this paper and demonstrated that the proposed framework is ideal for a real-world application.

V. CONCLUSION

Activity recognition is an important research area in the field of machine learning. Specifically in anomaly detection, approaches that need explicit labeling present challenges and generally are biased. Anomaly events are rare to occur, and there are no public datasets with real fall data, making more challenging to create solutions that can be applied in the real world. Thus, an approach that no needs labeled data to train a model is a good alternative in this scenario.

This paper proposed a deep anomaly detection framework for fall monitoring based on reconstruction, using an autoencoder through ST-GCN features. Our main objective was to create a robust solution that did not need knowledge of fall events during the training phase. Thus, the assumption was that training the model without artificial fall events would result in a better generalization.

Analyzing the obtained results, we observe that our proposed framework can identify fall events with a low number of false negative and false-positive events in most experiments. Furthermore, our framework can also detect other types of anomalies related to health risks, not limited to falls.

We encourage future work to extend the framework to recognize other health risk problems in the context of residences, making it more robust and applicable to monitoring people in residence without requiring a health care professional.

REFERENCES

[1] S. Chaudhuri, H. Thompson, and G. Demiris, "Fall detection devices and their use with older adults: A systematic review," *J. geriatric Phys. Therapy*, vol. 37, no. 4, p. 178, 2014.

[2] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *Biomed. Eng. OnLine*, vol. 12, no. 1, p. 66, 2013.

[3] J. Pynoos, B. A. Steinman, and A. Q. D. Nguyen, "Environmental assessment and modification as fall-prevention strategies for older adults," *Clinics Geriatric Med.*, vol. 26, no. 4, pp. 633–644, Nov. 2010.

[4] T. Xu, Y. Zhou, and J. Zhu, "New advances and challenges of fall detection systems: A survey," *Appl. Sci.*, vol. 8, no. 3, p. 418, Mar. 2018.

[5] P. Lameski, A. Dimitrievski, E. Zdravevski, V. Trajkovik, and S. Koceski, "Challenges in data collection in real-world environments for activity recognition," in *Proc. IEEE 18th Int. Conf. Smart Technol. (EUROCON)*, Jul. 2019, pp. 1–5.

[6] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, Dec. 2014.

[7] A.-C. P. I. M. B. Cramariuc. (2019). *Precis Har*. [Online]. Available: http://dx.doi.org/10.21227/mene-ck48

[8] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "UP-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, Apr. 2019.

[9] Y. Zheng, D. Zhang, L. Yang, and Z. Zhou, "Fall detection and recognition based on GCN and 2D pose," in *Proc. 6th Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2019, pp. 558–562.

[10] Y.-H. Nho, J. G. Lim, D.-E. Kim, and D.-S. Kwon, "User-adaptive fall detection for patients using wristband," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 480–486.

[11] Y. M. Galvao, V. A. Albuquerque, B. J. T. Fernandes, and M. J. S. Valenca, "Anomaly detection in smart houses: Monitoring elderly daily behavior for fall detecting," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI)*, Nov. 2017, pp. 1–6.

[12] L. Chen, R. Li, H. Zhang, L. Tian, and N. Chen, "Intelligent fall detection method based on accelerometer data from a wrist-worn smart watch," *Measurement*, vol. 140, pp. 215–226, Jul. 2019.

[13] M. M. Hassan, A. Gumaei, G. Aloi, G. Fortino, and M. Zhou, "A smartphone-enabled fall detection framework for elderly people in connected home healthcare," *IEEE Netw.*, vol. 33, no. 6, pp. 58–63, Nov. 2019.

[14] J.-S. Lee and H.-H. Tseng, "Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers," *IEEE Sensors J.*, vol. 19, no. 18, pp. 8293–8302, Sep. 2019.

[15] H. Jian and H. Chen, "A portable fall detection and alerting system based on k-NN algorithm and remote medicine," *China Commun.*, vol. 12, no. 4, pp. 23–31, Apr. 2015.

[16] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 1, pp. 156–167, Jan. 2006.

[17] H. Li, A. Shrestha, H. Heidari, J. L. Kernec, and F. Fioranelli, "Bi-LSTM network for multimodal continuous human activity recognition and fall detection," *IEEE Sensors J.*, vol. 20, no. 3, pp. 1191–1201, Feb. 2020.

[18] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Multiple cameras fall dataset," DIRO-Université de Montréal, Montreal, QC, Canada, Tech. Rep. 1350, 2010.

[19] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 1, pp. 314–323, Jan. 2019.

[20] K. Adhikari, H. Bouchachia, and H. Nait-Charif, "Activity recognition for indoor fall detection using convolutional neural network," in *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl. (MVA)*, May 2017, pp. 81–84.

[21] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 611–622, May 2011.

[22] X. Wang, L. Gao, J. Song, and H. Shen, "Beyond frame-level CNN: Saliency-aware 3-D CNN with LSTM for video action recognition," *IEEE Signal Process. Lett.*, vol. 24, no. 4, pp. 510–514, Apr. 2017.

[23] M. Yu, Y. Yu, A. Rhuma, S. M. R. Naqvi, L. Wang, and J. A. Chambers, "An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 6, pp. 1002–1014, Nov. 2013.

[24] S. Taghvaei and K. Kosuge, "HMM-based state classification of a user with a walking support system using visual PCA features," *Adv. Robot.*, vol. 28, no. 4, pp. 219–230, Feb. 2014.

[25] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, Oct. 2016.

[26] D. Zhao, S. Liu, D. Gu, X. Sun, L. Wang, Y. Wei, and H. Zhang, "Enhanced data-driven fault diagnosis for machines with small and unbalanced data based on variational auto-encoder," *Meas. Sci. Technol.*, vol. 31, no. 3, Mar. 2020, Art. no. 035004.

[27] M. T. García-Ordás, J. A. Benítez-Andrades, I. García-Rodríguez, C. Benavides, and H. Alaiz-Moretón, "Detecting respiratory pathologies using convolutional neural networks and variational autoencoders for unbalancing data," *Sensors*, vol. 20, no. 4, p. 1214, Feb. 2020.

[28] M. A. Al-Shabi, "Credit card fraud detection using autoencoder model in unbalanced datasets," *J. Adv. Math. Comput. Sci.*, vol. 33, no. 5, pp. 1–16, Aug. 2019.

[29] C. Zhao, X. Li, and H. Zhu, "Hyperspectral anomaly detection based on stacked denoising autoencoders," *J. Appl. Remote Sens.*, vol. 11, no. 4, 2017, Art. no. 042605.

[30] J. Nogas, S. S. Khan, and A. Mihailidis, "Fall detection from thermal camera using convolutional LSTM autoencoder," in *Proc. 2nd Workshop Aging, Rehabil. Independ. Assist. Living, IJCAI Workshop*, 2018.

[31] J. Nogas, S. S. Khan, and A. Mihailidis, "DeepFall: Non-invasive fall detection with deep spatio-temporal convolutional autoencoders," *J. Healthcare Informat. Res.*, vol. 4, no. 1, pp. 50–70, Mar. 2020.

[32] X. Cai, S. Li, X. Liu, and G. Han, "Vision-based fall detection with multi-task hourglass convolutional auto-encoder," *IEEE Access*, vol. 8, pp. 44493–44502, 2020.

[33] S. Vadivelu, S. Ganesan, O. V. R. Murthy, and A. Dhall, "Thermal imaging based elderly fall detection," in *Computer Vision—ACCV 2016 Workshops* (Lecture Notes in Computer Science), vol. 10118, C. S. Chen, J. Lu, and K. K. Ma, Eds. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-54526-4_40.

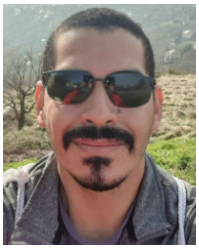
[34] X. Ma, H. Wang, B. Xue, M. Zhou, B. Ji, and Y. Li, "Depth-based human fall detection via shape features and improved extreme learning machine," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 6, pp. 1915–1922, Nov. 2014.

[35] H. Yhdego, J. Li, S. Morrison, M. Audette, C. Paolini, M. Sarkar, and H. Okhravi, "Towards musculoskeletal simulation-aware fall injury mitigation: Transfer learning with deep CNN for fall detection," in *Proc. Spring Simulation Conf. (SpringSim)*, May 2019, pp. 1–12.
[36] H. Sadreazami, M. Bolic, and S. Rajan, "TL-FALL: Contactless indoor fall detection using transfer learning from a pretrained model," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Jun. 2019, pp. 1–5.
[37] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. Van Gool, "Temporal 3D ConvNets: New architecture and transfer learning for video classification," 2017, *arXiv:1711.08200*. [Online]. Available: http://arxiv.org/abs/1711.08200
[38] C. Doersch and A. Zisserman, "Sim2real transfer learning for 3D human pose estimation: Motion to the rescue," 2019, *arXiv:1907.02499*. [Online]. Available: http://arxiv.org/abs/1907.02499
[39] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "ConvNet architecture search for spatiotemporal feature learning," 2017, *arXiv:1708.05038*. [Online]. Available: http://arxiv.org/abs/1708.05038
[40] Y. M. Galvão, J. Ferreira, V. A. Albuquerque, P. Barros, and B. J. T. Fernandes, "A multimodal approach using deep learning for fall detection," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114226. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417420309489
[41] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018, *arXiv:1801.07455*. [Online]. Available: http://arxiv.org/abs/1801.07455
[42] A. Abobakr, M. Hossny, and S. Nahavandi, "A skeleton-free fall detection system from depth images using random decision forest," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2994–3005, Sep. 2018.
[43] T. Xu and Y. Zhou, "Fall detection based on skeleton data," in *Human Aspects of IT for the Aged Population. Applications, Services and Contexts* (Lecture Notes in Computer Science), vol. 10298, J. Zhou and G. Salvendy, Eds. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-58536-9_38.
[44] T.-H. Tsai and C.-W. Hsu, "Implementation of fall detection system based on 3D skeleton for deep learning technique," *IEEE Access*, vol. 7, pp. 153049–153059, 2019.
[45] O. Keskes and R. Noumeir, "Vision-based fall detection using ST-GCN," *IEEE Access*, vol. 9, pp. 28224–28236, 2021.
[46] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*. [Online]. Available: http://arxiv.org/abs/1705.06950
[47] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 115–123.
[48] F. Harrou, N. Zerrouki, Y. Sun, and A. Houacine, "An integrated vision-based approach for efficient human fall detection in a home environment," *IEEE Access*, vol. 7, pp. 114966–114974, 2019.
[49] A.-C. Popescu, I. Mocanu, and B. Cramariuc, "Fusion mechanisms for human activity recognition using automated machine learning," *IEEE Access*, vol. 8, pp. 143996–144014, 2020.

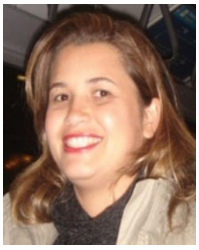
YVES M. GALVÃO received the bachelor's degree in information systems from the Faculdade Santa Maria, Brazil, and the master's degree in computer engineering from the University of Pernambuco (UPE), Brazil, where he is currently pursuing the Ph.D. degree in computer engineering. Furthermore, he has specialization in software engineering, and some relevant certifications in the same area, such as java enterprise architect, SOA architect, and PSM-II. His research interests include anomaly detection, applied mainly on fall detection problems, using deep learning, computer vision, and image recognition.

LETÍCIA PORTELA is currently pursuing the bachelor's degree in computer engineering with the University of Pernambuco, Recife, Brazil. Her research interests include computer vision, machine learning, and artificial intelligence. She has received the Academic Excellence Award and volunteered from the Academic Center of Computer Engineering.

JANDERSON FERREIRA received the bachelor's degree in computer science from FACAPE, Brazil, and the master's degree in computer engineering from UPE, Brazil, where he is currently pursuing the Ph.D. degree in computer engineering. Sandwich Period at the Universidad de Santiago de Compostela–Campus Santiago, Spain. He works with research and consultancy in the areas of computer vision, machine learning, and artificial intelligence.



PABLO BARROS received the Ph.D. degree in computer science from the University of Hamburg, Germany. He is currently a Research Scientist with the Italian Institute of Technology, Italy. He was the main Organizer of recent workshops and challenges at IJCAI, ICDL-EPIROB, IROS, IJCNN, and FG. His research interests include affective computing and affective robots, on the development of deep and self-supervised deep neural networks.



ROBERTA ANDRADE DE ARAÚJO FAGUNDES received the M.Sc. and Ph.D. degrees in computer science from the University Federal of Pernambuco (UFPE), in 2006 and 2013, respectively, and the post Ph.D. degree in statistic from UFPE, in 2015. She is currently an Adjunct Professor of computer engineering with the University of Pernambuco. She is also a Deputy Coordinator and a Teacher of the Post-Graduation Program in computer engineering (PPGEC).

BRUNO J. T. FERNANDES (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the Federal University of Pernambuco, Recife, Brazil, in 2007, 2009, and 2013, respectively. He is currently an Associate Professor with the University of Pernambuco. He is also CNPq Productivity Fellow in technological development; a Coordinator of the Graduate Program in computer engineering (master's and Ph.D.), UPE; a Coordinator of the Computer Vision Laboratory, Instituto de Inovação Tecnológica (IIT -UPE); and the Head of the Pattern Recognition and Digital Image Processing Research Group, UPE. His research interests include machine learning, computer vision, image processing, and neural networks. He was a recipient of awards, including the 2008 Google Academic Prize as the Top M.Sc. Student in the Federal University of Pernambuco and the Science and Technology Award for Outstanding Research in the Polytechnic School at the University of Pernambuco, in 2011 and 2017. He received the title of Livre-Docente from the University of Pernambuco in 2017.