Evaluation of fall detection approaches based on virtual devices
Leveraging on Motion Capture data in Unity environments

Eduarda Vaz[1], Heitor Cardoso[2] and Plinio Moreno[2]
[1]*Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal*
[2]*Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Torre Norte Piso 7, 1049-001 Lisboa, Portugal*
eduarda.vaz07@gmail.com,heitor.cardoso@tecnico.ulisboa.pt,plinio@isr.tecnico.ulisboa.pt

Keywords: Fall detection, wrist devices, sensor simulation, Unity environment.

Abstract: Realistic fall detection datasets are difficult to acquire due to the high risks, awkward situation of pretending to be falling and limited to young healthy individuals. In this work we propose to leverage on motion capture data acquired for games and animations, to simulate the recordings of accelerometers and orientation sensors. The simulated sensor values are obtained in the Unity environment. Our dataset allows to further evaluate the generalization properties of previously presented methods by including new types of both falling and non-falling samples. Our case study is the fall detection based on wristband devices.

1 INTRODUCTION

Average life expentancy and other cultural and social factors have increased the autonomous elderly population that live single(Fuster, 2017), bringing challenges on health systems because is very difficult to monitor and follow dangerous situations that have a long-term impact on health and independent living. Falls belong to those situations, which is the focus of this work. Recent developments on sensing devices and the small scale electronics have allowed the development of wearable gadgets that estimate biosignals for personal monitoring, which can be utilized by the patients and the healthcare personnel. We study the fast and accurate fall detection, using data provided by a 3D accelerometer and a gyroscope on the user's wrist.

Several works have addressed the fall detection as a classification problem, where the ideal algorithm cannot miss any fall ocurrence (i.e. Zero False Negative rate). Wearables on the hip (Sucerquia et al., 2017), chest (Torres et al., 2018), wrist (Khojasteh et al., 2018; de Quadros et al., 2018) and several body locations (Casilari et al., 2017b) have been designed and utilized for fall detection in daily activities. In this work we focus on the wristband devices and computationally efficient machine learning algorithms. Although several approaches have obtained good results using only accelerometers (Barri Khojasteh et al., 2018), the False Negative rate is very high for actual application of the algorithms. More recent approaches use inputs from an accelerometer, a gyroscope and a magnetometer, which aim to extract vertical components of acceleration and velocity (de Quadros et al., 2018). In this work we follow the same approach, using a 3D accelerometer and 3D orientation as sensory input for feature computation.

A robust detection of falls based on Machine learning approaches requires large amounts of data that covers all the possible daily activities as well as different types of realistic falls. Several datasets have been gathered in previous works (Sucerquia et al., 2017; Khojasteh et al., 2018; Casilari et al., 2017a), each one with a specific configuration of sensors and devices that limits the creation of a single large dataset. The most similar work that fits our wristband with 3D acceleration and 3D orientation is the one by (de Quadros et al., 2018), which contains six non-fall daily activities and four types of falls. We use these samples as the training set, and extend the experimental procedure by adding other activities and falls to the dataset. To extend this dataset, we leverage on Motion Capture data acquired for movies, games and animations[1] by creating a new fall dataset based on computer simulations in the Unity engine (Haas, 2014). Our new dataset includes people doing physical exercise, two new types of falls, more daily activities and daily activities performed by handicaped persons. We

[1] `https://mixamo.com`

evaluate the generalization and robustness of the models built with the dataset in (de Quadros et al., 2018). Figure 1 shows a field view of the Unity environment with 32 characters executing actions.

2 RELATED WORK

(Sucerquia et al., 2017) presents a dataset of falls and ADLs acquired with a device attached to the waist of participants, which incorporate acceleration and rotation data. The dataset contains 19 types of ADLs and 15 types of falls, performed by young adults and the elderly in a wide variety of activities. A threshold-based classifier was selected for this work, achieving up to 96% accuracy in detecting falls. However, validation testing with the elderly significantly reduced the fall detection performance of the features tested, as algorithms trained on data from young people tended to bias the thresholds upwards in amplitude. This type of result makes evident the need to include data from older people in the training phase.

Considering the future availability of multiple type of sensors in smart homes and buildings, (Martínez-Villaseñor et al., 2019) presents a dataset that considers RGB cameras, infrared sensors, and accelerometers and gyroscopes at several locations of the body of the person. The scenario is similar to the one of the SisFall dataset (Sucerquia et al., 2017), but adds on the multiple sensors in the environment. (Martínez-Villaseñor et al., 2019) shows that the additional information of the multi-modal sensors provides better fall detection results than using single-mode sensors. In a follow-up (Galvão et al., 2021) propose to detect the falls on the dataset by (Martínez-Villaseñor et al., 2019) using Neural Network architectures, which improve further the performance of fall detection. Although we aim to detect falls as well, the dataset of (Martínez-Villaseñor et al., 2019) and the fall detection of (Galvão et al., 2021), their application is limited to the smart-home environment, where the computational complexity needs to be moved to a centralized server. As stated in the introduction, our application scenario considers that the only available device is a wristband device for fall detection.

The work exposed in (Quadros et al., 2017; de Quadros et al., 2018) is the most recent on fall detection using only wrist devices, which is not a common configuration in the literature. For this protocol, twenty-two young adults were involved, repeating each activity three times. A total of twelve different activities were studied, where half of them are related to fall simulation and half simulate activities of daily life. After evaluating five different machine learning methods, the best result was presented by the k-NN method, resulting in 99% accuracy. In light of this result, it is shown that machine learning approaches with the proper motion decomposition are potentially capable of achieving optimal results for a fall detection system based on a wrist-worn device.

3 APPROACH

Fall detection is addressed as a binary classification problem, considering as raw features 3-dimensional acceleration and 3-dimensional orientation measured on the wrist. For every data recording, we compute magnitude of accelerations, velocities and displacements, averaged over fixed intervals of time. In addition, vertical components of acceleration, velocity and displacement are estimated using the orientation. Finally, for each recording, mean and maximum values of the (vertical) accelerations, velocities and displacements are arranged in various configurations as input features. Those input features are fed into the following classifiers: K-nearest neighbors (KNN) (Fix and Hodges, 1951), Linear Discriminant Analysis (LDA) (Fisher, 1936), Decision Trees (DT) (Breiman et al., 2017), Logistic Regression (LR) (Berkson, 1944) and Support-Vector Machine (SVM) (Cortes and Vapnik, 1995).

3.1 Feature computation

Raw acceleration values (X, Y and Z) are filtered using a median filter to reduce the noise. Then, we follow the same approach as in (de Quadros et al., 2018; Quadros et al., 2017) that at each sample, computes the average value of each acceleration component over one second. The basis of all the features is then the magnitude of the filtered and averaged acceleration (i.e. Total Acceleration, TA). Finally, for each sequence the mean and maximum TA values are the basic features for each sequence to be classified.

Based on the TA values of a sequence, velocity and displacement features are computed through integration (i.e. Total Velocity, TV; and Total Displacement, TD). Similar to TA-based features, mean and maximum values are obtained for each sequence.

The 3D orientation provides the sensor orientation with respect to the earth, where usually the Z component (i.e. vertical component) corresponds to the majority of the acceleration during fall occurrences. In addition to TA, TV and TD, we obtain the vertical component of these features, adding the Vertical Acceleration (VA), Vertical Velocity (VV) and Vertical

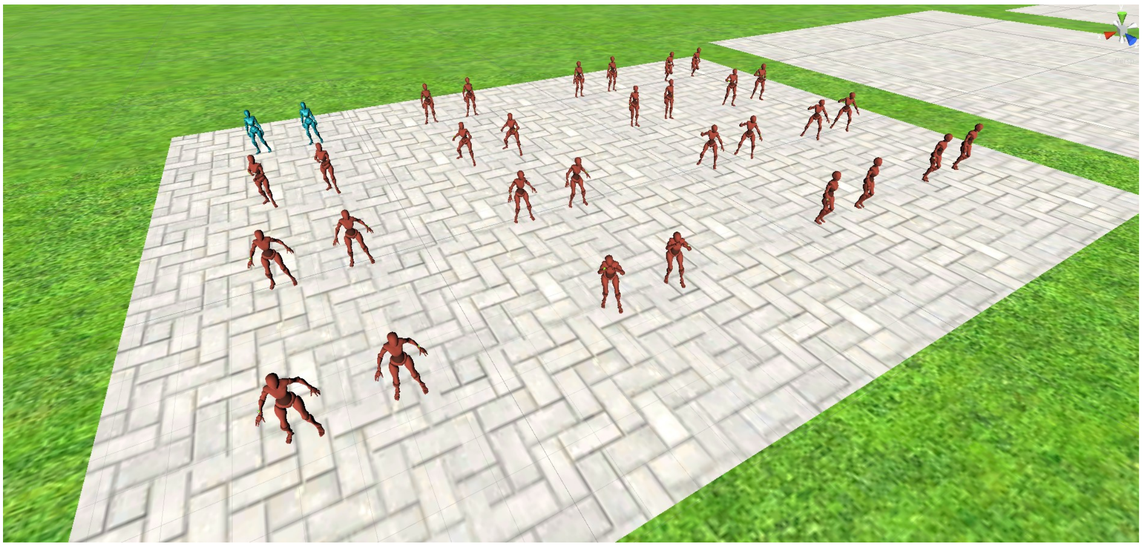


Figure 1: Visualization of the Unity environment of our fall dataset. In this example, each character executes the motion of a type of fall activity.

Displacement (VD) to the set of available features. The final set of features includes: TA, TV, TD, VA, VV and VD.

3.2 Classifiers

Since the feature space is low dimensional and the number of samples on the (Quadros et al., 2017; de Quadros et al., 2018) dataset is 792 (Note that just 22 people executed the motions), we resort to conventional machine learning algorithms that have good learning rates and work robustly on these conditions (Adadi, 2021), such as K-nearest neighbors, Decision Trees and Logistic Regression. We also consider Linear Discriminant Analysis and Support-Vector Machine.

4 DATASETS

The wrist-based device dataset by (de Quadros et al., 2018; Quadros et al., 2017) is our training dataset, which we use to create the classification model. We contribute with a new testing dataset, which leverages from freely available full-body character animations captured from professional motion actors [2]. We add several simulated sensors on the wrist: Accelerometer, Gyroscope and Orientation, running the simulations in Unity to generate the samples for the testing dataset in this work.

4.1 Arduino-based device

The dataset by (Quadros et al., 2017) designs a wristband based on an GY-80 device[3] connected to an Arduino UNO, which is connected by cable to a desktop computer (see Figure 5). The data recorded during the exectution of the activities includes 3D accelerations, 3D rotational velocities and 3D magnetic field information. The dataset protocol summarized in Table 1, considers Activities of Daily Life (ADL) in the non-fall class, and four types of fall.

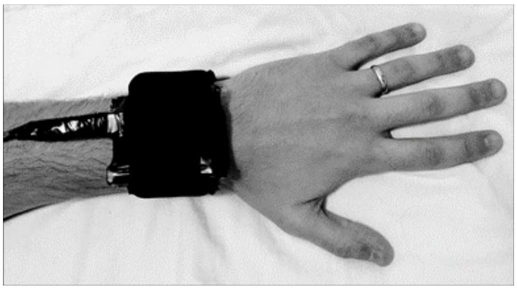


Figure 2: Handcrafted wrist device designed in (Quadros et al., 2017).

The dataset contains various types of falls and a reduced set of non-fall activities (ADLs), which are oriented towards the elderly use-case scenario. Falls such as trip-based falls and syncope[4] falls are not considered in this dataset. In addition, other non-fall activities such workout exercises, gestures and idle are not included in this dataset. We want to evaluate the

[2] https://mixamo.com

[3] Referred to as Inertial Measurment Unit (IMU)

[4] Fainting due to cardiovascular abnormalities

Table 1: Summary of activities and number of samples per class of the (Quadros et al., 2017) dataset. Twenty two subjects participated in the recordings and each subject performed every activity three times, which yields 792 samples.

Class	Activity
Fall 18 samples/person	Forward fall
	Backward fall
	Sideways to device's side
	Sideways to no-device side
	Fall after waist clockwise rotation
	Fall after waist counterclockwise rotation
Non-fall 18 samples/person	Walking
	Clapping
	Open and close door
	Moving object
	Tying a shoe
	Sitting on chair

generalization properties of Machine learning models obtained from (Quadros et al., 2017) dataset to other types of falls and daily activities.

4.2 Virtual device in Unity

We develop an Unity environment that includes several characters from Mixamo. For each character, a predefined motion as listed in Table 2 is executed. While the motion is executed we compute the motion statistics that emulates the following sensors on the wrist: (i) 3D accelerometer, (ii) 3D gyroscope and (iii) 3D orientation.

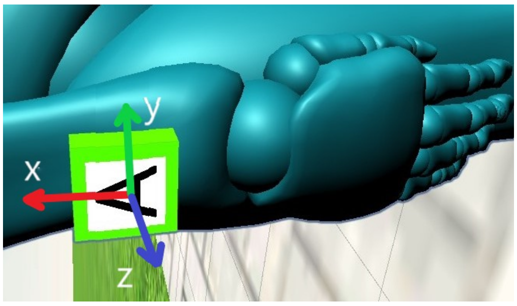


Figure 3: Reference frame of the sensor computations in Unity. Note that Unity uses Left hand coordinate systems.

During the execution of the animation, Unity provides the pose (i.e. position $\mathbf{x}$ and orientation matrix $\mathbf{A}$) of the wrist with respect to the world frame. We use backward differences to compute the linear velocity ($\mathbf{v}$), linear acceleration ($\mathbf{a}$) and angular velocity (ω):

$$\mathbf{v}(t) = \frac{\Delta \mathbf{x}(t)}{\Delta t} \tag{1}$$

$$\mathbf{a}(t) = \frac{\Delta \mathbf{v}(t)}{\Delta t} \tag{2}$$

$$\omega(t) = \frac{\Delta \mathbf{A}}{\Delta t}\mathbf{A}. \tag{3}$$

The finite differences in Eqs. (1-3) provide the sensor values for each animated character.

Regarding the falling class, we consider two new types of falls: Trip and syncope. Pictures from the animated characters for these type of falls are shown in Figure 4. Regarding the non-falling class, we consider:

- Workout exercises: Air squats, burpee, running, jumping jacks, Frisbee throw, dribble
- Gestures: pointing, praying, waving, writing, handshake
- Idle: Bored balancing while stand, idle stand, idle sit, idle talking on the phone, idle laying
- Walking: Walking with a walker, walking forward, walking backward, walking and turn 180, walking injured
- Daily living exercises: Crouch to stand, stairs ascending and descending, standing up

Table 2 summarizes the Unity dataset classes and its corresponding activities per class. The dataset is parsed into a JSON (Pezoa et al., 2016) file that contains the simulated sensor values and their corresponding activity type and class label for all the 162 samples.

Table 2: Summary of activities and number of samples per class of the Unity dataset. LH denotes the character wears the device on the Left Hand and RH on the Right Hand. Total number of samples is 162.

Class	Activity	Hand
Fall 32 samples	Trip fall	11 LH, 11 RH
	Syncope fall	5 LH, 5 RH
Non-fall 130 samples	Workout exercises	13 LH, 13 RH
	Daily Living exercises	6 LH, 6 RH
	Gestures	11 LH, 11 RH
	Walking	15 LH, 15 RH
	Idle	20 LH, 20 RH

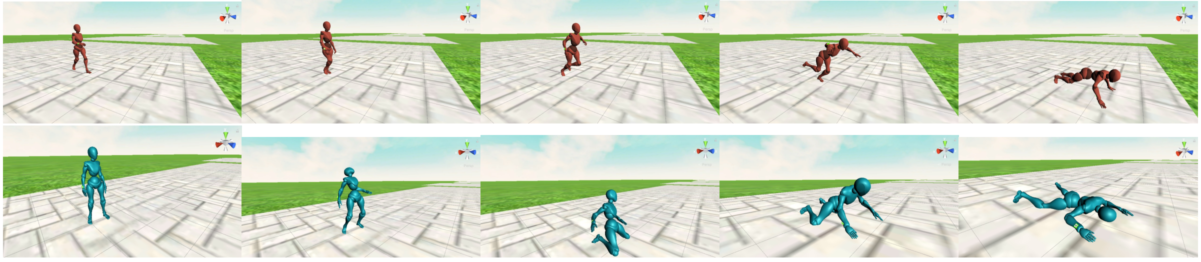
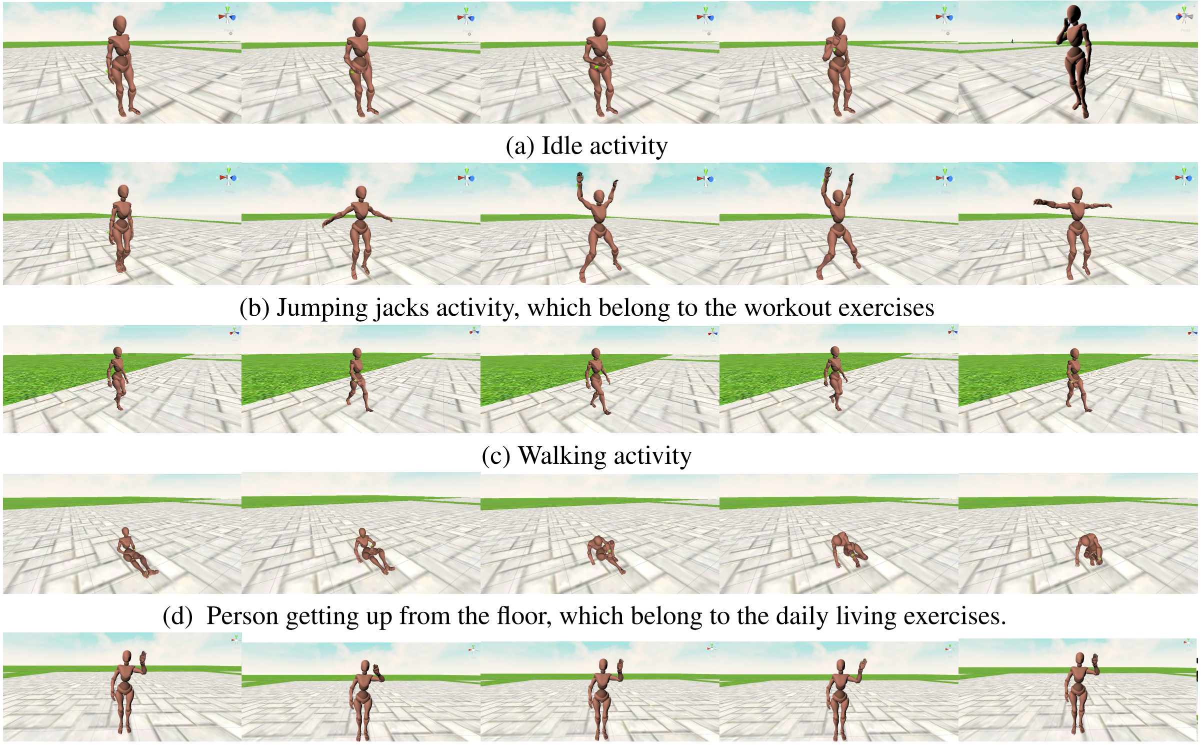


Figure 4: Visualization of the sequence of frames of unity fall types. The top row shows the character during a trip fall and the bottom row shows the character during the syncope fall.



(a) Idle activity

(b) Jumping jacks activity, which belong to the workout exercises

(c) Walking activity

(d) Person getting up from the floor, which belong to the daily living exercises.

(d) Waving hand activity, which belong to the gestures.

Figure 5: Visualization of the sequence of frames of unity non-fall activities.

5 EXPERIMENTS

We evaluate the generalization capabilities of the classification models by (de Quadros et al., 2018), using as testing set our animated characters dataset in Unity. The evaluation measures include accuracy, sensitivity (i.e. true positive rate) and specificity (i.e. true negative rate). In the case of fall detection the ideal target is not missing the occurrence of a fall event (i.e. highest sensitivity). If we have competing algorithms with similar accuracy, we prefer the approach with higher sensitivity. The feature sets and classifiers corresponds to the ones described in (de Quadros et al., 2018), which were selected by testing several feature sets. The feature sets with best results for each classifier are summarized in Table 3.

Table 3: Feature sets and classifiers from (de Quadros et al., 2018).

Feature set	Classifier
VA, VV, VD	k-NN
TA, VA, TV	LDA
VA, TV	LR
VA, TV	DT
VA, TV	SVM

We follow the k-fold cross-validation method to evaluate the deviations from the average values of accuracy, sensitivity and specificity as shown in Table 4. The best model is the DT classifier with Vertical

Acceleration and Total Velocity as features, considering its high mean accuracy and sensitivity values while having a low deviation from the corresponding means. Following DT, we have SVM and LDA with similar accuracy values, but SVM has a higher sensitivity.

Table 4: Evaluation of different machine learning methods for (de Quadros et al., 2018) dataset. Sensitivity (Sens.), Specificity (Spec.) and Accuracy (Acc.) results.

	Sens. (%)	Spec. (%)	Acc. (%)
k-NN	92,66 ± 4,03	84,32 ± 7,06	88,02 ± 4,60
LDA	91,68 ± 4,33	93,94 ± 3,42	92,05 ± 3,41
LR	89,04 ± 4,17	93,48 ± 3,11	90,65 ± 2,62
DT	**92,93 ± 3,35**	93,16 ± 3,65	**93,06 ± 2,13**
SVM	93,91 ± 4,28	90,39 ± 4,56	92,05 ± 2,93

The models that provide top accuracy values for each one of the experimental setups in Table 3 are selected to classify the samples in our new character simulated dataset. The best classifier is k-NN with VA, VV and VD features, because reaches perfect sensitivity while having the second better accuracy.

Table 5: Results after applying the machine learning models obtain by training (de Quadros et al., 2018) dataset in Unity dataset. Sensitivity (Sens.), Specificity (Spec.) and Accuracy (Acc.) results.

	Sens. (%)	Spec. (%)	Acc. (%)
k-NN	**100**	64,62	71,61
LDA	40,63	63,08	58,64
LR	90,63	35,39	46,30
DT	53,13	**77,69**	**72,84**
SVM	68,75	66,15	66,66

6 CONCLUSIONS AND FUTURE WORK

In this work we present an extension of a fall detection dataset, where users are wearing wristband devices/smartwatches. The new dataset leverages on Motion Capture data acquired for movies, games and animations, which are inserted in the Unity engine to simulate the wristband sensors. Compared to the dataset in (Quadros et al., 2017), we include two new falling types and a large variety of non-falling activities such as workout exercises, gestures and idle ones. This large set of non-falling samples serves to evaluate the generalization capabilities of the models developed by (de Quadros et al., 2018). The k-NN model is able to detect all the fall events in our new dataset, but its true negative is low. To address this issue, future work must develop a larger Unity dataset that should be merged with the dataset in (de Quadros et al., 2018) in order to create models with better generalization.

ACKNOWLEDGEMENTS

This publication has been partially funded by the project LARSyS - FCT Project UIDB/50009/2020 and the project and by the project IntelligentCare – Intelligent Multimorbidity Managment System (Reference LISBOA-01-0247-FEDER-045948), which is co-financed by the ERDF – European Regional Develpment Fund through the Lisbon Portugal Regional Operational Program – LISBOA 2020 and by the Portuguese Foundation for Science and Technology – FCT under CMU Portugal.

REFERENCES

Adadi, A. (2021). A survey on data-efficient algorithms in big data era. *Journal of Big Data*, 8(1):1–54.

Barri Khojasteh, S., Villar, J. R., de la Cal, E., González, V. M., Sedano, J., and Yazgan, H. R. (2018). Evaluation of a wrist-based wearable fall detection method. In de Cos Juez, F. J., Villar, J. R., de la Cal, E. A., Herrero, Á., Quintián, H., Sáez, J. A., and Corchado, E., editors, *Hybrid Artificial Intelligent Systems*, pages 377–386, Cham. Springer International Publishing.

Berkson, J. (1944). Application of the logistic function to bio-assay. *Journal of the American statistical association*, 39(227):357–365.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Casilari, E., Santoyo-Ramón, J.-A., and Cano-García, J.-M. (2017a). Analysis of public datasets for wearable fall detection systems. *Sensors*, 17(7).

Casilari, E., Santoyo-Ramón, J. A., and Cano-García, J. M. (2017b). Umafall: A multisensor dataset for the research on automatic fall detection. *Procedia Computer Science*, 110:32–39. 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017) / 12th International Conference on Future Networks and Communications (FNC 2017) / Affiliated Workshops.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

de Quadros, T., Lazzaretti, A. E., and Schneider, F. K. (2018). A movement decomposition and machine learning-based fall detection system using wrist wearable device. *IEEE Sensors Journal*, 18(12):5082–5089.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Fix, E. and Hodges, J. J. (1951). Discriminatory analysis. nonparametric discrimination: Consistency properties. Technical report, University of California, Berkeley.

Fuster, V. (2017). Changing demographics: a new approach to global health care due to the aging population.

Galvão, Y. M., Ferreira, J., Albuquerque, V. A., Barros, P., and Fernandes, B. J. (2021). A multimodal approach using deep learning for fall detection. *Expert Systems with Applications*, 168:114226.

Haas, J. K. (2014). A history of the unity game engine.

Khojasteh, S. B., Villar, J. R., Chira, C., González, V. M., and De la Cal, E. (2018). Improving fall detection using an on-wrist wearable accelerometer. *Sensors*, 18(5).

Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., and Peñafort-Asturiano, C. (2019). Up-fall detection dataset: A multimodal approach. *Sensors*, 19(9):1988.

Pezoa, F., Reutter, J. L., Suarez, F., Ugarte, M., and Vrgoč, D. (2016). Foundations of json schema. In *Proceedings of the 25th International Conference on World Wide Web*, pages 263–273. International World Wide Web Conferences Steering Committee.

Quadros, T. d. et al. (2017). Development and evaluation of an elderly fall detection system based on a wearable device located at wrist. Master's thesis, Universidade Tecnológica Federal do Paraná.

Sucerquia, A., López, J. D., and Vargas-Bonilla, J. F. (2017). Sisfall: A fall and movement dataset. *Sensors*, 17(1).

Torres, G. G., Bayan Henriques, R. V., Pereira, C. E., and Müller, I. (2018). An enocean wearable device with fall detection algorithm integrated with a smart home system. *IFAC-PapersOnLine*, 51(10):9–14. 3rd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control CESCIT 2018.