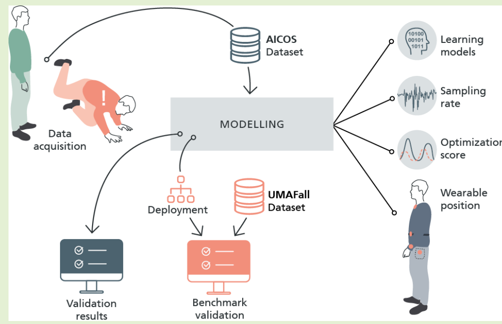
Automated Development of Custom Fall Detectors: Position, Model and Rate Impact in Performance

Joana Silva, Diana Gomes, Inês Sousa, and Jaime S. Cardoso, *Senior Member, IEEE*

***Abstract*—The past years have witnessed a boost in fall detection-related research works, disclosing an extensive number of methodologies built upon similar principles but addressing particular use-cases. These use-cases frequently motivate algorithm fine-tuning, making the modelling stage a time and effort consuming process. This work contributes towards understanding the impact of several of the most frequent requirements for wearable-based fall detection solutions in their performance (usage positions, learning model, rate). We introduce a new machine learning pipeline, trained with a proprietary dataset, with a customisable modelling stage which enabled the assessment of performance over each combination of custom parameters. Finally, we benchmark a model deployed by our framework using the UMAFall dataset, achieving state-of-the-art results with an F1-score of 84.6% for the classification of the entire dataset, which included an unseen usage position (ankle), considering a sampling rate of 10 Hz and a Random Forest classifier.**



***Index Terms*—Fall detection, inertial sensors, accelerometer, machine learning, deep learning.**

I. INTRODUCTION

FALL detection systems have been a trend research topic over the past years, motivated by the damaging impact of fall events in the quality of life, especially of the elder, and the importance of prompt assistance to minimize their consequences. Among the variety of available solutions, wearable-based systems, relying on ubiquitous equipment (e.g. smartphone, smartwatch, fitness trackers) to enable pervasive monitoring of users' motion parameters, are some of the most common. As such, there is a tendency to generate multiple fall detection solutions adapted to each different use case and shaped by each system's hardware limitations. This leads to an overflow of custom-made systems built upon similar methodologies but fine-tuned to particular objectives, constraints or even target populations.

Common examples of specific requirements and constrains are related to the wearable design, such as the place of usage, the way it can be attached to the body; the device's processing capability, memory and battery; or limitations in the accelerometer sampling rate. Fall detection systems' fine-tuning implies the collection of a significant amount of data examples, in conditions as similar as possible to those of the intended use, to train and test a new fall detection model. Hardware specifications may also influence the choice of the modelling approach and adaptations in the implementation of the model may be required. In summary, adjusting multiple fall detection solutions is a time and effort consuming process.

In this work, we introduce a new machine learning pipeline, trained with data from a comprehensive proprietary dataset, to model and deploy custom-made fall detection algorithms, based on which we shall:

1) Study the cases in which customization is indeed necessary

- *Model complexity:* Do models of higher complexity outperform models of more modest complexity at detecting falls?
- *Sensor position generalization:* Do models that were not trained with data from sensors placed on a

Manuscript received January 6, 2020; revised January 30, 2020; accepted January 30, 2020. Date of publication February 3, 2020; date of current version April 16, 2020. This work was supported in part by the Project Indoor Activity Notification for Vigilance Services, funded under the AAL JP and co-funded by the European Commission under Grant AAL-2018-5-116 and in part by the National Funding Authorities of Portugal, Belgium, and Switzerland. The associate editor coordinating the review of this article and approving it for publication was Prof. Kea-Tiong Tang. *(Corresponding author: Joana Silva.)*

Joana Silva is with Fraunhofer Portugal AICOS, 4200-135 Porto, Portugal, and also with the Faculty of Engineering, University of Porto, 4099-002 Porto, Portugal (e-mail: joana.silva@fraunhofer.pt).

Diana Gomes and Inês Sousa are with Fraunhofer Portugal AICOS, 4200-135 Porto, Portugal (e-mail: diana.gomes@fraunhofer.pt; ines.sousa@fraunhofer.pt).

Jaime S. Cardoso is with INESC TEC, 4200-465 Porto, Portugal, and also with the Faculty of Engineering, University of Porto, 4099-002 Porto, Portugal (e-mail: jaime.cardoso@inesctec.pt).

Digital Object Identifier 10.1109/JSEN.2020.2970994

certain body position maintain their performance when evaluated with these data?
- *Single vs. multiple training positions:* Do models solely trained with data from sensors placed on a certain body position A perform better than models trained with data acquired at multiple positions when evaluated with data from A?
- *Sampling rate:* Does the accelerometer sampling rate have an impact in fall detection performance?

2) Evaluate the performance of our framework against the state-of-the-art
- *External data generalization:* Do models deployed by our framework perform adequately at detecting falls using datasets acquired under different conditions?
- *Positioning within state-of-the-art:* Is the performance of a model deployed by our framework competitive within the state-of-the-art?

All in all, this study makes significant contributions towards: i) understanding if customization is indeed necessary for a specific use case, namely regarding usage position, accelerometer sampling rate, and processing/performance trade-off requirements; ii) the automated creation of mature ready-to-go fall detection solutions adapted to several of the most frequent customisation requirements for wearable-based systems.

This manuscript is organised as follows: section II presents related works; section III discloses our methodology; section IV presents the obtained results; section V discusses those results, comparing them with related state-of-the-art; section VI closes the document, presenting its most relevant conclusions.

II. Related Work

According to a recent review of Ren *et. al* [1] of fall detection systems, the taxonomy of these systems can be divided into context-based and wearable-based systems. Context-based systems can sense and process data from the environment where the sensing device is integrated, instead of using a device attached to the person. Examples of context-based systems are pressure platforms, cameras, acoustic and infrared sensors, as described by Chaccour *et. al* [2]. Most wearable-based systems resort to the analysis of inertial sensors to detect falls. However, there are also systems that rely on sensor fusion, ranging from the fusion of inertial sensors to a combination of these sensors with barometer, microphone, heart rate sensors or cameras. Wang *et al.* [3], for example, combined accelerometer and barometric information in a fall detector with low-power consumption.

Regardless of the data source, the most common data analysis algorithms in the state-of-the-art can be divided into three main groups: threshold-based algorithms, binary or multiclass machine learning supervised algorithms, and one class classification or novelty detection algorithms. The threshold-based approaches are simple algorithms that trigger a fall alarm when the sensor values exceed certain predefined thresholds or a set of rules. Contrarily, machine learning approaches based on pattern recognition are more complex and sophisticated compared to threshold-based approaches. In novelty detection algorithms only data from daily life movements are used for training, and falls are detected as outliers.

Fall detection systems are usually trained and evaluated in simulated scenarios, given the difficulty of acquiring data from real-world falls. There are also research studies that attempted to collect data from real falls in uncontrolled settings; however, fall events are very rare and its respective number of samples is frequently insufficient to train robust supervised models, as in the two datasets used in the work of Aziz *et al.* [4]. Most previous works have acquired data from simulated falls and ADLs. Besides acquiring scripted samples in laboratory conditions, other studies have focused on acquiring and evaluating the trained models in free-living scenarios, from a continuous usage of the wearable devices, like Alves *et al.* [5]. Nevertheless, fall detection is usually an unbalanced problem, with a higher percentage of non-falls compared to falls in most datasets reported in prior works.

The system setup depends on several variables that could influence fall detection performance, and there has been some effort in previous studies towards understanding the impact of the wearable device usage position and the sensors' sampling rate. Wearable position can influence the type of movements that could be misinterpreted as a fall, e.g. trunk, waist and pocket positions are expected to trigger fewer false alarms than the wrist, given its higher number of degrees of freedom, as in Ozdemir *et al.* [6] work. Santoyo-Ramón *et al.* [7] investigated the impact of number and positions of wearable sensors in fall detection. Their findings suggest that the best usage positions for the wearable devices are the chest and/or waist. On the other hand, sampling rate has an impact in computational efficiency and battery life of the system. Liu *et al.* [8] studied this topic and tested several models with lower sampling rates, and the SVM obtained 97% accuracy, with a sampling rate of 5.8 Hz. In this sense, position and rate are both important to consider at the design stage, and assume an important role in the scope of this work.

We will benchmark our method with the publicly available UMAFall dataset from Casilari *et al.* [9]. Thus, we have surveyed state-of-the-art works which used this dataset for evaluation of their own methodology. Tsinganos *et al.* [10] have extracted features from the accelerometer magnitude, considering only the belt position. These features were used to train a k-NN classifier. Their validation method was not user-independent (because they did not use leave-one-subject-out (LOSO) validation), and achieved a F1-score of 96.7%. The work of Wisesa *et al.* [11] revealed a F1-score of 97.4% using a LSTM model solely trained with data from the X-axis of the accelerometer from the belt position. For validation of results, the authors have divided the dataset into two static parts at random, disregarding user-independence; thus, the obtained results are not only orientation-dependent, but may also be optimistic if aiming real-world utilisation with unseen users. The work of Khojasteh *et al.* [12] compared a decision tree (DT) model with a feed-forward neural network (NN). The authors have

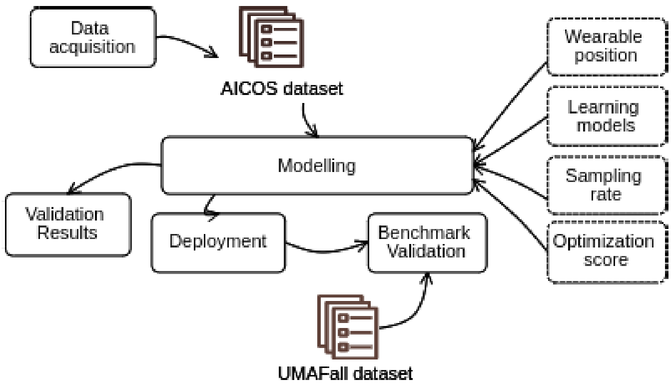


Fig. 1. Study design overview.

validated their models considering only the wrist position and applying a 5x2 cross validation, which can also be considered user-dependent. The DT slightly outperformed the NN model regarding the geometric mean of sensitivity and specificity (DT - 92.4%; NN - 91.8%). The work of Wang *et al.* [13] was the only study found with evaluation with the UMA dataset using LOSO cross validation. Their best approach combines data from accelerometer and gyroscope in a threshold-based algorithm. The highest obtained results were 95.3% sensitivity and 81.5% specificity (i.e, 88% geometric mean), although the authors did not refer if all UMA dataset positions were considered to evaluate their results.

III. Methods

Figure 1 depicts an overview of the proposed approach for automated development of custom fall detectors, enabling a clearer understanding of the relation between each stage within the flow of the method. The following subsections detail the steps at each of these stages.

A. Data Acquisition

1) Protocol: Fraunhofer AICOS has been acquiring simulated falls and non-falls since 2009. The protocol for data collection was first described by Aguiar *et al.* [5], [14], which followed the protocol defined by Noury *et al.* [15], and considers data collection for the on-body sensor positions of chest, belt, and pocket. Recently, that protocol was extended to include the wrist position and non-fall movements specific to the wrist. The dataset was collected in laboratory conditions, at AICOS' living lab, where two mattresses were placed in the ground. The living lab also included a sofa, a table with chairs, a bed and an open space for acquiring running and walking samples. The activities of daily living recorded as non-falls included drop the sensor on the table, sit on a lower chair, catch an object from the floor while walking, run a few meters, laying on a bed, among others. The type of falls recorded included forward, backward and lateral falls (without recovery) ending lying on the floor. The protocol was previously described by Aguiar *et al.* [14] and Alves *et al.* [5]. Overall, the dataset comprises 36 different types of falls and 43 types of non-falls. Data was collected using a data logger Android application that provides access to the inertial sensors either directly built-in the smartphone or in wearable devices paired with the smartphone. The wearable devices used are proprietary of Fraunhofer AICOS and include a 3-axis Inertial Measurement Unit (IMU) [16]. Several smartphone models were used for data collection, namely: Samsung S3, S3 Mini, S4, Nexus S, Galaxy Nexus, Nexus 5, Moto G XT1032, and Vodafone 985N.

TABLE I
Distribution of Dataset Across Different Positions in Terms of Number of Subjects, Fall and Non-Fall Samples

Position	Subjects	Fall	Non-fall	*Total*
Belt	24	1731	1407	3138
Pocket	28	1305	1146	2451
Wrist	7	887	1112	1999
Chest	12	455	401	856
Total	42 (unique)	4378	4066	8444

2) Data Distribution: Data was collected in several occasions, from different participating subjects who wore a set of devices in different on-body locations. For this reason, none of the subjects has collected data for the complete set of usage positions considered in this study. For each subject, the positions for which only one class is available (fall or non-fall) were removed prior to the analysis. The cleaned dataset is composed by 42 subjects (34 male) with average age of 25.0 $\pm$ 2.9 years, an average weight of 72.4 $\pm$ 12.6 kg, and an average height of 176.0 $\pm$ 7.9 cm. The percentage of samples that were captured by the built-in sensors of the smartphones was 54.17% and the percentage acquired with the wearable devices was 45.81%. The average sampling rate for the smartphone samples was 102.26 $\pm$ 24.11 Hz and for the wearable samples the average sampling rate was 97.68 $\pm$ 8.50 Hz. The accelerometer range was $\pm$ 2G for all used smartphone models and wearable devices. The distribution of samples across the two classes is presented in Table I. The belt and pocket sensor positions have a higher percentage of samples than chest and wrist positions, because belt and pocket positions include samples from the smartphone and from wearable devices, whereas the chest and wrist include only samples from wearable devices. Overall, the distribution of falls and non-falls per position for the entire dataset may be considered nearly balanced. On average the fall events have a duration of 15.20 $\pm$ 4.99 seconds and the ADLs activities have a duration of 14.94 $\pm$ 5.30 seconds.

B. Modelling

Figure 2 illustrates the pipeline for automated modelling, using the AICOS dataset. This pipeline is prepared to receive several input parameters which enable the customized modelling (see Figure 1): 1) train and test positions; 2) learning models; 3) target sampling rate; 4) grid-search optimization score. In the scope of this work, all experiments were performed using the F1-score as the optimization score.

1) Data Pre-Processing: A resampling strategy was firstly implemented with the aim of correcting the time distribution of all arriving samples and compensating for eventual sensor reading gaps. The accelerometer signal magnitude was evenly sampled, according to the target sampling rate. To that end, we computed the expected time of arrival of each sample (t_e).

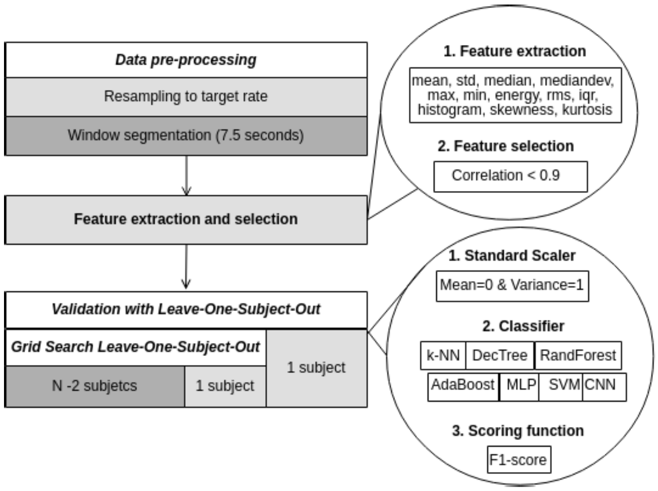


Fig. 2. Modelling stage: data pre-processing, feature extraction and selection, and nested leave-one-subject-out validation with grid search.

TABLE II
DIFFERENT COMBINATIONS OF INPUT PARAMETERS TESTED USING THE MODELLING PIPELINE

Type of test	Positions Train	Positions Test	Target rate (Hz)	Learning model
Baseline	P, B, C, W	P, B, C, W	100	All
Unseen test position	C, B, W	P	100	Random Forest
	P, B, W	C	100	
	P, C, W	B	100	
	P, C, B	W	100	
Single position	P	P	100	Random Forest
	C	C	100	
	B	B	100	
	W	W	100	
Rate variation	P, B, C, W	P, B, C, W	50	Random Forest
	P, B, C, W	P, B, C, W	20	
	P, B, C, W	P, B, C, W	10	
	P, B, C, W	P, B, C, W	5	
	P, B, C, W	P, B, C, W	3	
	P, B, C, W	P, B, C, W	1	

P–Pocket; B–Belt; C–Chest; W–Wrist

Samples arriving before t_e were stacked and their average was computed and set to correspond to t_e; if there were no samples arriving before t_e, the value of the last sample which arrived in the stream was considered. This combination of up and down-sampling techniques resulted in the computation of the accelerometer signal magnitude, evenly distributed in time, according to the required sampling rate. The data stream was segmented into windows of 7.5 seconds, without overlap, centered in the signal magnitude maximum. If there were not enough samples in the beginning or in the end of the window, after centering it in the maximum, the first and/or the last samples, respectively, were replicated until the pre-defined window size is reached. Windows with a standard deviation of low accelerometer magnitude were removed in order to discard samples that were useless for training the fall detection algorithm.

2) Features: Several time-domain features were extracted for each time-window signal magnitude: mean, standard deviation, median, median deviation, maximum, minimum, energy, root mean square, inter quartile range, histogram (10 bins), skewness and kurtosis, using our open source Time Series Feature Extraction Library [17]. These features require low computation power and are the most common used features for fall detection according to Pannurat *et al.* [18]. Features with correlation higher than 0.90 were removed. All features of the training set were standardized by removing the mean and scaling to unit variance. The same parameters were used to standardize the test set. These features constituted the input for all classifiers, with the exception of CNN. CNN received a feature vector of raw signal magnitude (for each time-window), re-scaled to $[0, 1]$ range by subtracting the minimum and dividing by the difference between maximum and minimum signal magnitude.

3) Leave-One-Subject-Out Validation: Two nested LOSO loops were used for training and validation assessment. The inner LOSO was used to optimize the hyperparameters of the learning models via grid search (except for the CNN-1D model) using N-2 participants for training and 1 subject for validation.

- *Grid search for hyperparameters optimization:* The hyperparameters of the learning models were optimized for F1-score metric. The following hyperparameters were optimized for each classifier: k-Nearest Neighbours (k-NN), parameter k and search algorithm; Decision Tree (DT) & Random Forest (RF), maximum depth, number of features and estimators, and minimum samples to split, AdaBoost, number of estimators; Multi-layer Perceptron (MLP), variable alpha, activation function and learning rate; Support Vector Machine (SVM), variable C, degree, gamma and type of kernel.
- *CNN-1D architecture:* the architecture of the network encompasses two stacked 1-Dimensional Convolutional Neural Network (CNN-1D) with kernel size of 5, with 4 filters, and tangent activation function. CNN-1D layers were interleaved with max pooling and 0.25 dropout layers. The sigmoid function was used in the last activation layer. The loss function was set to the binary cross-entropy and optimized with Adam algorithm.

The outer LOSO was used to assess the performance of the best set of parameters, retrieved from the grid search (inner LOSO), in the remaining subject of the dataset. The final output metrics, presented in section IV, were computed by mapping correct and misclassifications by user, position and learning model. This process enabled the computation of single (cumulative) confusion matrices with respect to each of these parameters, from which all performance metrics were extracted: accuracy (Acc), sensitivity (Se), specificity (Sp), precision (Prec), F1-score (F1), Youden index (YI), and geometric mean of sensitivity and specificity (G). As such, this outer LOSO was paramount to enable the fair comparison of algorithms defined by different input parameters, maintaining complete user-independence in the validation process.

C. Multiple Comparisons

ANOVA multiple comparison analysis was used for comparison of performance metrics between different tests, using vectors of metrics by user obtained from the outer LOSO validation loop as input. As *post-hoc* test, we used the Tukey's Honest Significant Difference test (95% confidence level) between pairs of different learning models, usage positions or sampling rates. These tests aimed the identification of statistically significant differences between different combinations of input parameters (Table II), in order to address the research questions of this work.

All learning models were considered and compared pair-wise for training and testing with all positions at 100 Hz (*Baseline*). For simplicity of analysis, we selected a single model - Random Forest - for conducting all remaining tests, based on the results of the aforementioned comparison and the fact that it is a decision-based classifier. Algorithms based on decision trees are very interpretable, do not require much computation, and are ease to implement in any platform. A more detailed explanation of this selection process is provided in subsection IV-A.

D. Deployment

The output metrics of the LOSO validation in the modelling stage shall assist the process of selecting the most adequate learning model for deployment, considering the requirements of each specific use case, i.e. the selection process should consider performance, complexity and/or other requirements initially setup for the algorithm.

After the selection of the classification algorithm, all data of the AICOS dataset corresponding to the required positions (and resampled to the desired target rate) are used to refit the classifier, with the respective best set of hyperparameters derived from the process of LOSO grid search. This step completes the deployment of a final fall detection algorithm.

To evaluate the effectiveness of our method, we have deployed a fall detector algorithm using a Random Forest classifier, expecting a sampling rate of 10 Hz, and trained with all positions available in the AICOS dataset. This algorithm was then tested using all data from the UMAFall dataset for performance comparison with other fall detection works using the same data.

E. Benchmark Validation Using the UMAFall Dataset

We benchmarked our framework with the publicly available UMAFall dataset [19] described in Casilari *et al.* [7], [9]. Several ADLs and simulated falls were collected from 17 volunteers with an average age of 26.7 ± 10.5 years old. Each subject wore four different wearable devices – chest, belt, wrist, ankle -- and carried one smartphone in the pocket. Overall, 11 types of ADLs and 3 types of falls were simulated, yielding a total of 970 falls and 2444 non-falls, with an average of 683 samples for each usage position. Accelerometer, gyroscope and magnetometer data was collected at a sampling rate of 20 Hz from the wearables and 200 Hz from the smartphone.

The UMAFall dataset was selected for its representation of all sensor positions included in AICOS dataset. Interestingly, it also contains data from wearables positioned in a new position -- the ankle –, which our framework is not expecting, and shall thus allow us to assess the generalization of the deployed fall detector for this new usage position.

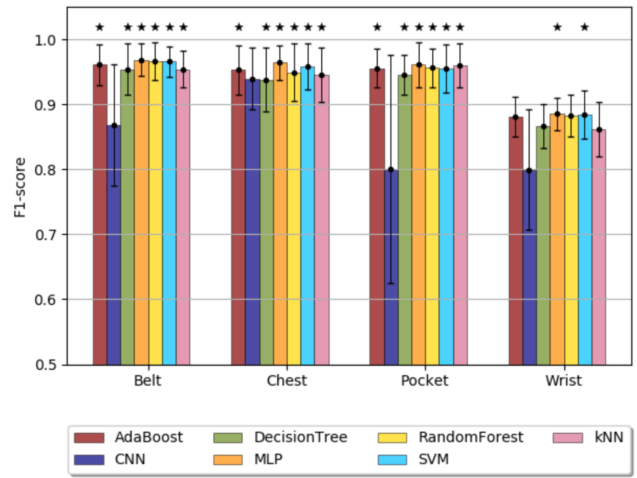


Fig. 3. F1-score for all tested classifiers, considering the baseline input parameters. Classifiers with SSD from CNN for each sensor position are marked with stars.

IV. Results

A. Multiple Comparisons

Even though we analysed multiple comparisons for several performance metrics, we opted for solely presenting the results for the F1-score for simplicity of analysis, since it was selected as the scoring metric in the optimization process. Moreover, the F1-score will allow us to assess the performance of the algorithm taking into account an harmonic mean of precision and recall.

The first set of comparisons corresponded to the performance of different learning models for the same set of input parameters (defined as *Baseline* in Table II). Results were arranged by position and classifier and exhibited in Figure 3. We looked for statistically significant differences (SSD) between all pairwise combinations of classifiers. No SSD were found among the conventional supervised binary models tested within each position; however, CNN's performance was frequently significantly inferior to that of the remaining models. Given the equivalence of all the conventional models tested, all subsequent experiments were performed using a single classification model. We prioritised decision-based models (Decision Tree and Random Forest) in this selection, due to their low prediction expensiveness which is valuable for wearable implementations. Random Forest was finally selected since it consistently led to higher average F1-scores than Decision Tree classifiers.

Figure 4 presents the results for different combinations of train/test sensor positions, organised by sensor position of test data. Multiple comparison analysis was performed between results for respective positions derived from setting as input parameters: 1) *Baseline* vs. *Unseen test position*; 2) *Baseline* vs. *Single position*. No SSD were found between either of them. This means that, for example, for 1), the performance of detecting falls in data from sensors in the pocket remains unchanged irrespective of whether data from sensors in this position are included in the training set or not; and, for 2), solely using data from sensors in the pocket for training does not improve the performance of fall detection in data from

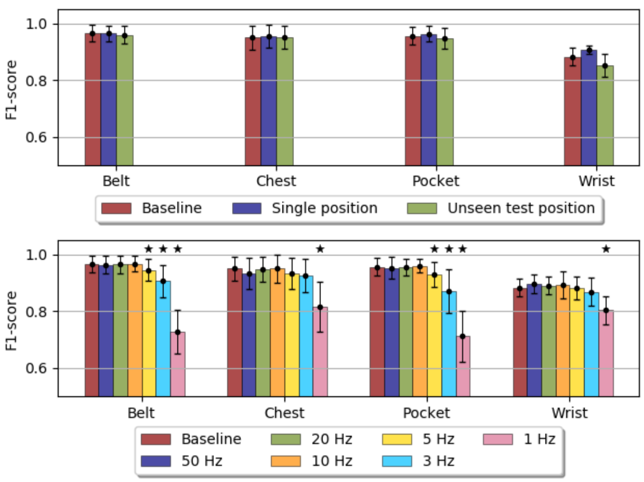


Fig. 4. F1-score for Random Forest classification, considering the described combinations of input parameters. Pipelines with SSD from *Baseline* for each sensor position are marked with stars.

TABLE III
EVALUATION RESULTS WITH THE UMAFALL DATASET. ALL PERFORMANCE METRICS ARE IN %

Position	F1	Acc	Se	Sp	Prec	YI	G	N
Belt (B)	95.0	96.9	95.2	97.7	94.7	92.9	96.4	617
Pocket (P)	90.2	94.2	92.8	94.7	87.7	87.5	93.7	723
Chest (C)	90.8	94.9	87.5	97.9	94.3	85.4	92.5	725
Wrist (W)	81.5	87.3	97.1	83.4	70.1	80.5	90.0	726
Ankle (A)	65.3	78.9	78.3	79.1	55.9	57.5	78.7	623
PBCW	88.8	93.2	93.1	93.2	84.9	86.3	93.2	2786
PBCWA	84.6	90.6	90.7	90.5	79.2	81.2	90.6	3414

F1–F1-score; Acc–Accuracy; Se–Sensitivity; Sp–Specificity; Prec–Precision; YI–Younden Index; G–Geometric mean; N–No. Samples

sensors in the pocket, relatively to including data from all the different sensor positions in the training set.

Finally, fall detection performance results using data sampled at different rates are depicted in Figure 4. A Random Forest classifier was trained and tested using data from sensors in all the positions available in AICOS dataset and varying the accelerometer sampling rate (*Baseline* and *Rate variation* entries of Table II). Considering rates of 100 Hz, 50 Hz, 20 Hz or 10 Hz did not lead to SSD between the fall detection performance for respective test positions. However, statistically significant decays of performance were verified for belt and pocket positions for rates of 5 Hz and 3 Hz, and, more evidently, for all positions with data sampled at 1 Hz.

B. Benchmark Validation

Table III combines the results obtained using the framework's model specifically deployed for benchmark validation, as previously described, organised by testing data sensor positions, for all sensor positions included in AICOS dataset (i.e. except ankle), and for the entire dataset. All computed metrics were presented for analysis to instigate further comparisons with previous and future works in the field.

The belt sensor position presented, overall, the best results, immediately followed by pocket and chest positions - the first associated with more false positive occurrences (lower specificity) and the latter associated with more false negative occurrences (lower sensitivity). For data from sensors placed on the wrist a decrease of performance was verified, as compared with the previous positions, which is coherent with the results obtained using the AICOS dataset (Figure 3). Finally, considering the testing data from sensors on the ankle yielded the poorest performance for all compared metrics. Combining the samples of all positions, we achieved an F1-score of 84.6%, which increased until 88.8% by not considering the unexpected ankle position.

V. DISCUSSION

This section will provide an overall discussion of results, considering general results, CNNs compared with standard techniques, and impact of positions and sampling rate in the performance of the models. Moreover, a state-of-the-art performance comparison will be presented along with potential limitations of this study.

A. Need for Customization

Figures 3 and 4 provide important information towards understanding the cases worthy of investment in customization.

Starting with the problem of selecting the most adequate learning model, considering trade-offs of performance and available resources in wearable implementations, one can take the results depicted in Figure 3, which unveiled that there is no SSD between the performance of all standard binary classification models in our tests for all considered sensor positions. If we describe model complexity as a function of its consumed resources and prediction expensiveness, one can observe that *there is no evidence that higher model complexity leads to improved results* in the conditions under which these tests took place. This means that selecting the least complex model for implementation may be beneficial for the final system, because it shall lead to lower resource consumption while achieving statistically similar results. If this conclusion is taken under consideration at the moment of system design, there may not be a need to develop several fall detectors with custom learning models to improve performance under different restrictions on the availability of resources.

Figure 4 enables a discussion of the role of considering (or not) sole data from the intended place of usage of the sensing device in the training stage. Our tests verified that the *fall detection performance on data from each of the* 4 *sensor positions is similar, regardless of its inclusion in the training stage*, using AICOS dataset. While this conclusion is not particularly surprising for belt and pocket (both at the waist), or even chest (all in the trunk region), to achieve similar performance for the wrist regardless of its consideration in the training stage is not intuitive. This conclusion can reiterate a claim for position generalization of our method, even though further tests should be conducted to thoroughly understand if there is a more significant impact for other performance metrics. Moreover, to *solely consider data from the intended sensor position to train the models leads to statistically similar results than considering all positions as training data*. As such, it may be beneficial to consider all positions at the modelling stage, regardless of the effective place of usage of the final system, so that its portability is facilitated under different conditions, if needed.

From the rate impact study, one can conclude that the *lowest sampling rate considered that did not present SSD from the baseline 100 Hz pipeline was 10 Hz*. This conclusion appears to be coherent with findings of previous works [8], setting a valuable landmark in the field of fall detection towards the efficiency of wearable systems.

B. State-of-the-Art Performance

The quality of the AICOS dataset, regarding its variety of usage positions, the representative amount of samples for each position, and expression of relevant different types of falls and non-falls, allowed us to deploy a robust Random Forest classifier trained with all usage positions of this dataset for a target rate of 10 Hz, since no SSD were found between these models and models trained considering higher sampling rates. This process based the conclusion that *our framework is able to deploy models that perform adequately when tested with data acquired under different conditions* (not controlled by the authors), as Table III corroborates.

The authors of the UMA dataset have achieved their best results for chest and belt [7], comparing with other usage positions, consistently with our findings, to which we can add the pocket position in our case (performance similar to chest and belt). Moreover, the geometric mean achieved in that work was always inferior to 75% for any combination that included the sensor in the ankle, which means that even though our dataset did not feature any sample acquired from the ankle position, our method still outperforms the method of the authors of the dataset at detecting falls in this position (79% geometric mean for the sole classification of ankle samples).

Directly comparing previous studies with our user-independent approach is, however, a difficult task, since the validation methods previously reported are mostly user-dependent; thus, it is unclear if these methods would lead to the same results under user-independent conditions, typically more challenging. The work of Wang *et al.* [13] was the only study found employing LOSO cross validation. Comparing that work with ours, one can verify that our method achieved better results (geometric mean of 91% *vs.* 88%) for all positions considered in the UMA dataset. However, the authors did not explicitly refer if all UMA dataset positions were considered to evaluate their results.

It is also worth mentioning that the model that we have deployed was trained with data downsampled to 10 Hz, instead of using the most frequent sampling rate of the UMA dataset, 20 Hz. In spite of that, *the results obtained with our models are in line with those of other studies using the same data*.

C. Limitations

These conclusions may not be true for all datasets, but only for datasets similar to AICOS; they are maybe only true due to the quality of our dataset, and the higher amount of samples for each usage position, that allowed us to generalize better to new unseen positions. Moreover, these results were obtained using the F1-score as the optimization score. One can also analyse all of the pipelines' comparisons for other scoring metrics, and the conclusions found with the F1-score may not stand. The model deployed by our framework retrieved from the pipeline described in this study should also be validated with more datasets, and ideally with data from real fall events.

VI. Conclusion

In this work, we studied the impact of learning models, on-body positioning and sampling rate in fall detection performance, using a new machine learning pipeline which is able to deploy fall detection solutions adapted to the aforementioned system requirements. Our experiments did not verify any relation between model complexity and performance. Moreover, using our dataset and method, considering 3 positions in the training set was enough for achieving model generalization for the 4th (unseen) position, and considering solely data from a certain position *vs.* all positions in the training stage led to statistically similar results when detecting falls at that position. We were also able to decrease the sampling rate expected by our pipeline until 10 Hz without any statistically significant impact in performance.

Finally, we used the UMAFall dataset to benchmark a solution deployed by our framework. This solution is expected to receive data sampled at 10 Hz and uses a Random Forest classifier previously trained with data from AICOS dataset. This experiment unveiled that our solution led to state-of-the-art results for the UMAFall dataset, even under our demanding test conditions (considering an unseen test position, the ankle; lower sampling rate; test data acquired under conditions not controlled by the authors).

As future work, we can optimize our pipeline for different performance metrics (other than F1-score), to deploy models that require a specific trade-off between sensitivity and specificity. For example, in a specific case or disease it can be more important to detect falls than to have a higher rate of false alarms. This framework will ease the fast deployment of fall detection models that are adjusted to different use cases. After selecting the most suitable model and the target performance metric, we expect to implement our pipeline in a wearable solution to assess the model's performance in free-living conditions.

Acknowledgment

The authors would like to thank all the participants involved in the collection of the AICOS dataset, and the authors of the UMAFall dataset.

References

[1] L. Ren and Y. Peng, "Research of fall detection and fall prevention technologies: A systematic review," *IEEE Access*, vol. 7, pp. 77702–77722, 2019.

[2] K. Chaccour, R. Darazi, A. H. El Hassani, and E. Andres, "From fall detection to fall prevention: A generic classification of fall-related systems," *IEEE Sensors J.*, vol. 17, no. 3, pp. 812–822, Feb. 2017.

[3] C. Wang *et al.*, "Low-power fall detector using triaxial accelerometry and barometric pressure sensing," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2302–2311, Dec. 2016.

[4] O. Aziz *et al.*, "Validation of accuracy of SVM-based fall detection system using real-world fall and non-fall datasets," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0180318.

[5] J. Alves, J. Silva, E. Grifo, C. Resende, and I. Sousa, "Wearable embedded intelligence for detection of falls independently of on-body location," *Sensors*, vol. 19, no. 11, p. 2426, May 2019.

[6] A. Özdemir, "An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice," *Sensors*, vol. 16, no. 8, p. 1161, Jul. 2016.

[7] J. Santoyo-Ramón, E. Casilari, and J. Cano-García, "Analysis of a smartphone-based architecture with multiple mobility sensors for fall detection with supervised learning," *Sensors*, vol. 18, no. 4, p. 1155, Apr. 2018.

[8] K.-C. Liu, C.-Y. Hsieh, S. J.-P. Hsu, and C.-T. Chan, "Impact of sampling rate on wearable-based fall detection systems based on machine learning models," *IEEE Sensors J.*, vol. 18, no. 23, pp. 9882–9890, Dec. 2018.

[9] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "UMAFall: A multisensor dataset for the research on automatic fall detection," *Procedia Comput. Sci.*, vol. 110, pp. 32–39, Jan. 2017.

[10] P. Tsinganos and A. Skodras, "On the comparison of wearable sensor data fusion to a single sensor machine learning technique in fall detection," *Sensors*, vol. 18, no. 2, p. 592, Feb. 2018.

[11] I. W. W. Wisesa and G. Mahardika, "Fall detection algorithm based on accelerometer and gyroscope sensor data using recurrent neural networks," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 258, May 2019, Art. no. 012035.

[12] S. B. Khojasteh, J. R. Villar, E. de la Cal, V. M. González, and C. Chira, "Comparing model performances applied to fall detection," in *Proc. Int. Conf. Math. Appl.*, 2018, pp. 1–6.

[13] F.-T. Wang, H.-L. Chan, M.-H. Hsu, C.-K. Lin, P.-K. Chao, and Y.-J. Chang, "Threshold-based fall detection using a hybrid of tri-axial accelerometer and gyroscope," *Physiol. Meas.*, vol. 39, no. 10, Sep. 2018, Art. no. 105002.

[14] B. Aguiar, T. Rocha, J. Silva, and I. Sousa, "Accelerometer-based fall detection for smartphones," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Jun. 2014, pp. 1–6.

[15] N. Noury *et al.*, "Fall detection-principles and methods," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2007, pp. 1663–1666.

[16] "A day with pandlets," Fraunhofer AICOS, Porto, Portugal, White Paper, 2016.

[17] Fraunhofer AICOS. (2019). *TSFEL: Time Series Feature Extraction Library*. [Online]. Available: https://github.com/fraunhoferportugal/tsfel

[18] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: A review," *Sensors*, vol. 14, no. 7, pp. 12900–12936, Jul. 2014.

[19] E. Casilari, J. Santoyo-Ramón, and J. Cano-García. (2018). *UMAFall: Fall Detection Dataset*. [Online]. Available: https://doi.org/10.6084/m9.figshare.4214283.v7

Joana Silva received the master's degree in bioengineering from the Faculty of Engineering, University of Porto, in 2013. During the master's, she has worked in projects related to physical activity monitoring and walking data analysis using wearable devices, and her master's thesis focused in the area of physical activity monitoring using smartphones. She is currently pursuing the Ph.D. degree in machine learning applied to fall prediction and detection using wearable sensors. Since 2013, she has been a Researcher with Fraunhofer Portugal AICOS in the area of falls and activity monitoring. The work developed under the master's thesis was reward with the Young Scientist Best Paper Award in the p-Health Conference 2014 in Vienna.

Diana Gomes received the M.Sc. degree in bioengineering, in the field of biomedical engineering, from the Faculty of Engineering, University of Porto, in 2017. She is currently a Researcher with Fraunhofer Portugal AICOS, where she has been conducting research activities since 2017. During her studies, she was involved in several projects, including an internship at AGH-UST, Krakow, Poland, related to physiological signals processing, computer vision, and machine learning. Since the development of her master's thesis, she has focused her research in the field of human activity recognition through inertial sensing and signal processing, namely developing assistive technology for the elder and sports performance monitoring solutions.

Inês Sousa received the M.Sc. and Ph.D. degrees in biomedical engineering from the Instituto Superior Técnico, University of Lisbon, in 2007 and 2013, respectively. Her Ph.D. thesis focused on functional magnetic resonance imaging of the brain using quantitative methods and was developed in collaboration with Siemens Healthcare. She is currently the Head of Intelligent Systems and a Senior Scientist with Fraunhofer Portugal AICOS, where she has been researching on machine learning, signal processing, and time series analysis, specifically focusing on human motion analysis based on inertial sensors.

Jaime S. Cardoso (Senior Member, IEEE) received the Licenciatura (5 year) degree in electrical and computer engineering, the M.Sc. degree in mathematical engineering, and the Ph.D. degree in computer vision from the University of Porto in 1999, 2005, and 2006, respectively. He is currently an Associate Professor with Habilitation with the Faculty of Engineering, University of Porto, where he has been teaching machine learning and computer vision in Ph.D. Programs and multiple courses for the graduate studies. He is also the Coordinator of the Centre for Telecommunications and Multimedia, INESC TEC. His research can be summed up in three major topics computer vision, machine learning, and decision support systems. From 2012 to 2015, he served as the President of the Portuguese Association for Pattern Recognition (APRP), affiliated in the International Association for Pattern Recognition (IAPR).