Title: Three-layer Activity Recognition Combining Domain Knowledge and Meta-classification

Author list: Simon Kozina¹, Hristijan Gjoreski¹, Matjaž Gams¹, Mitja Luštrek¹

Affiliation of authors:

¹ Department of Intelligent Systems, Jožef Stefan Institute, Ljubljana, Slovenia

Running title: Three-layer Activity Recognition

*Corresponding author: Simon Kozina

Tel: +386 1 477 3195
Fax: +386 1 477 3131
E-mail: simon.kozina@ijs.si
ABSTRACT

One of the essential tasks of healthcare and smart-living systems is to recognize the current activity of a particular user. Such activity recognition (AR) is demanding when only limited sensors are used, such as accelerometers. Given a small number of accelerometers, intelligent AR systems often use simple architectures, either general or specific for their AR. In this paper, a novel system for AR, named TriLAR, is presented. TriLAR has an AR-specific architecture consisting of three layers: (i) a bottom layer, where an arbitrary number of AR methods can be used to recognize the current activity; (ii) a middle layer, where the predictions from the bottom-layer methods are inputs for a hierarchical structure combining domain knowledge and meta-classification; and (iii) a top layer, where a hidden Markov model is used to correct spurious transitions between the recognized activities from the middle layer. The middle layer itself has a hierarchical, three-level structure. First, a meta-classification is used to make the initial separation between the most distinct activities. Second, domain knowledge in the form of rules is used to differentiate between the remaining activities, recognizing the ones that it is designed for (i.e., static activities). Thirdly, another meta-classifier deals with the remaining activities. In this way, each activity is recognized by the method best suited to it, leaving unrecognized activities to the next method. This architecture was tested on a dataset recorded by ten volunteers performing a complex, real-life scenario while wearing accelerometers placed on the chest, thigh and ankle. The results showed that TriLAR successfully recognized elementary activities using one or two sensors and significantly outperformed three, standard, single-layer methods with all sensor placements.

Keywords: Activity recognition, Ambient intelligence, Intelligent healthcare, Machine learning, Meta-classification, Multi-layer activity recognition.
1 Introduction

The world’s population is aging rapidly, threatening to overwhelm society’s capacity to take care of its elderly members. The percentage of persons aged 65+ in the developed countries is projected to rise from 7.5 % in 2009 to 16 % in 2050 [1]. This is driving the development of innovative healthcare and smart-living technologies to help the elderly live independently for longer and with minimal support from the working-age population [2,3].

To be used in a real-world setting, healthcare and smart-living systems must understand the user’s situation and context, making activity recognition (AR) an essential component of such systems [4,5]. AR requires a sensor system that observes the user and intelligent software that infers the user’s activities from the sensor data [6,7].

The idea for our AR method was initiated and gradually developed in two European healthcare projects: Confidence [8] and CHIRON [9]. Even though AR was not the main goal in any of the projects, it eventually emerged as one of the most important components, being the foundation for further reasoning in the main tasks – the detection of falls, the detection of unusual behavior, the estimation of human energy expenditure and others [10-12]. The initial AR model developed in the Confidence project was a traditional, single-layer classification model that used two types of sensors (accelerometers and location sensors) to achieve adequate performance [13]. In the CHIRON project, the AR model was upgraded to two layers, which increased the performance.

In this paper a novel, three-layer architecture for the AR, i.e., TriLAR, is presented. The TriLAR architecture consists of the following: a bottom layer (an arbitrary number of independent AR methods), a middle layer (hierarchical aggregation that aggregates the predictions from the bottom-layer methods), and a top layer (a hidden Markov model that uses the temporal dependence of activities to remove the spurious transitions between them). This architecture was
tested on a dataset recorded by ten volunteers performing a complex, 90-minute scenario while wearing accelerometers placed on the chest, thigh and ankle. The results showed that by using the TriLAR architecture it was possible to successfully recognize elementary activities using a minimum number of sensors, i.e., one or two.

The paper is organized as follows. Firstly, an overview of the related studies on AR is presented in Section 2. In Section 3, the methods used in the TriLAR are described. Section 4 describes the experimental setup, including the sensor equipment, the experimental data and the methods setup. Section 5 includes the results and discussion. Finally, we draw conclusions and give directions for future work in Section 6.

2 Related Work

AR approaches can be divided into those using non-wearable sensors and those using the wearable type. The most common non-wearable approach is based on cameras [14]. Although this approach is physically less intrusive for the user compared to the wearable sensors, it suffers from problems such as low image resolution, target occlusion and time-consuming processing. However, often the biggest issue is user privacy: the user has to accept the fact that a camera will record him/her.

The most exploited and probably the most mature approach to AR is using wearable accelerometers, which are both inexpensive and effective [15-17]. This is also the reason why wearable accelerometers were used for our TriLAR architecture. There are two common types of wearable-sensor approaches to AR that have proved to be successful: using domain knowledge encoded with rules, and using machine learning (ML). Most researchers use a single-layer architecture, i.e., implementing only one of the two approaches. However, in recent years some more advanced, multi-layer, hierarchical approaches have also been proposed [18], some of them
exploiting the temporal dependence between human activities [19,20]. Further related work is divided into four paragraphs, each dedicated to one of the approaches.

The most traditional AR approach is that based on ML. This approach has a single-layer architecture and usually implements known classification methods, e.g., decision trees, SVM, kNN, Naive Bayes, etc. Examples include Kwapisz et al. [15], who used an accelerometer placed on the thigh and tested their single-layer approach by comparing the results of three classification methods on dynamic activities such as walking, running and jogging. However, for the elderly, static activities are also of great importance, as these activities are the main indicators of any degradation in their health (e.g., an increase in the time spent lying down). Ravi et al. [21] used an accelerometer on a mobile phone and tested their single-layer approach with five classification methods. The results showed that when the same person's data was used for training and testing, the accuracy was 90%, but when a different person's data was used for the testing, the accuracy dropped to 65%. In the TriLAR architecture, such classification methods are used on the bottom layer and the evaluation is performed on the data from different people, as the developed model is intended for use by people who were not involved in the training of the model.

Another common approach to accelerometer AR is based on manually created rules. These rules are usually based on features that are calculated from sensor orientations and accelerations. Wu et al. [16] presented an approach in which decision rules are used to recognize activities. Even though the rules are only one of several components in their approach, they showed that rules can successfully contribute to the AR. Another implementation of such rules is presented by Lai et al. [17]. The authors used six accelerometers placed on the neck, waist, left wrist, right wrist, left thigh, and right thigh. The reported accuracy was almost perfect, i.e., 99.5%, but the number
of sensors is excessive for everyday use. In the TriLAR architecture the domain rules are one of several methods used for the bottom and middle layers. In addition, we show that the TriLAR architecture achieves adequate performance when only using one or two sensors.

Hierarchical approaches implementing ensemble of classifiers to recognize a user's activity have been a popular research topic in the recent years. Banos et al. [18] showed that using traditional aggregation techniques such as majority vote [22,23] is not sufficient for a highly accurate AR system. For this reason, they presented a hierarchical-weighted classification (HWC), which is combination of the majority vote and weighted hierarchical aggregation. In their implementation, at the first level each sensor makes decision about the recognized activity using binary classifiers. At the next level, weighted majority vote scheme aggregates the decision in order to make the final decision. In our approach, each bottom-layer method decides about the user's activity using all the information available (from all sensors). Afterwards, the middle layer of the TriLAR architecture aggregates the outputs of the bottom layer using meta-classification techniques and domain rules.

Finally, because human activities have certain natural regularities and temporal dependence (smoothness), e.g., people do not abruptly switch back and forth between lying and cycling, the history of recent activities can help in recognizing the current activity. A common way to address this problem, and consequently reduce spurious activity transitions, is by using hidden Markov models (HMMs) [24]. Lester et al. [20] showed that incorporating HMMs significantly improves the recognition of activities. In the TriLAR architecture the HMM is used on the top layer, after the aggregation on the middle layer.
3 Methods

An overview of an AR system using the TriLAR architecture is shown in Figure 1. Data from wearable sensors is preprocessed and fed to the bottom layer, and the user’s activity is returned by the top layer.

In the bottom layer of the TriLAR architecture, three AR methods are used to recognize the current activity of a user. The first method uses the domain knowledge encoded with rules (rule-based AR), the second method uses trained classification models (binary classification) and the third method uses similarity metric (distance) between the activities in order to recognize the current activity. The methods are fundamentally different, so that each has an advantage in different situations. In the next layer, the middle layer, a hierarchical scheme aggregates the predictions from the bottom-layer classification methods and makes a joint decision about the user's activity. Because the traditional aggregation techniques, such as majority vote, did not achieve adequate results, a hierarchical, three-level, aggregation structure was designed that incorporates both ML (meta-classification) and domain knowledge (rules). In the top layer, a HMM incorporates the temporal component of the human activity, and corrects the final decision about the recognized activity. Each of the layers, as well as the preprocessing, is explained in more detail in the subsections that follow.

3.1 Data Preprocessing

The sensor equipment used in this study consists of three 3-axis accelerometers placed on the chest, thigh and ankle. These accelerometers are able to measure the accelerations in three directions (axes). The acceleration in each direction is the sum of the acceleration due to gravity and the acceleration due to the movement of the sensor.
The first step in the preprocessing phase is sensor-data synchronization. This is necessary when multiple sensors are used, since the data from all the sensors is not received at the same time. Once the sensor measurements are synchronized, further pre-processing is performed using a band-pass and a low-pass filter. The band-pass filter has two goals: (1) to eliminate the low-frequency acceleration (gravity) that captures information about the orientation of the sensor with respect to the ground and (2) to eliminate the high-frequency signal components generated by non-human motion and high-frequency noise, thus preserving the medium-frequency signal components generated by dynamic human motion. The band-pass filtered data is used for the extraction of features relevant for dynamic activities, such as walking, running and cycling. The low-pass filter has the opposite purpose: to eliminate most of the signal generated by dynamic human motion and preserve the low-frequency component, i.e., gravity [25,26]. In this way the low-pass filtered data contains the sensor-orientation information, which is relevant for the recognition of the static activities (postures), such as lying, sitting, standing and kneeling.

Finally, an overlapping sliding-window technique is applied. A window of fixed size (width) moves across the stream of data, advancing by half its length in each step. The data within each window is used in the AR described in the next section.

### 3.2 TriLAR Architecture

#### 3.2.1 Bottom-layer Methods

In the TriLAR's bottom-layer, three methods were implemented, each representing a distinct approach to AR. The first method is based on rules that recognize the posture using domain knowledge. In the second method, a set of binary classifiers is trained. Each classifier is trained to distinguish only one activity from the rest. As this increases the specificity of the classifiers, the accuracy should also increase. The last method involved the use of the k-nearest neighbors
As the majority of datasets are annotated manually, the data can contain spurious annotations. The kNNs should reduce the influence of such data. In general, though, the bottom layer can consist of an arbitrary number of AR methods.

Domain-knowledge rules use the low-pass filtered sensor orientations as the input, while the binary classifiers and the kNNs use a longer feature vector as the input. This also contains low-pass filtered features that measure the posture of the body. Additionally, it contains band-pass filtered features that represent: (1) the motion shape, (2) the motion variance, (3) the motion energy, and (4) the motion periodicity [25]. The feature vector consists of a total of 60 features per sensor.

### 3.2.1.2 Rule-based Activity Recognition

Rule-based activity recognition (R-BAR) is used for detecting the static activities, such as standing, lying and sitting. The dynamic activities, like walking or running, are merged with their static equivalent, standing. R-BAR uses the orientation of the sensors to recognize the posture. The orientation of a sensor $j$ is computed with Eq. (1), where $i$ is one of the axes $x, y$ or $z$. The orientation is simultaneously normalized to the $[0,1]$ interval.

$$
\phi_{j,i} = \left( \arccos \left( \frac{a_i}{\sqrt{a_x^2 + a_y^2 + a_z^2}} \right) + 1 \right) \cdot \frac{1}{2}
$$

The values computed in this way form an orientation vector $O=(\phi_{j,x}, \phi_{j,y}, \phi_{j,z})^{\text{number of sensors}}$, which is then matched with the set of rules defined by a domain expert as the typical orientations of the sensors for each activity. Figure 2 shows the example orientations for three activities (sitting, on all fours and standing) when using chest and thigh sensors. The structure of the rules in Figure 2 is $O_{activity}=(\phi_{\text{chest,x}}, \phi_{\text{chest,y}}, \phi_{\text{chest,z}}, \phi_{\text{high,x}}, \phi_{\text{high,y}}, \phi_{\text{high,z}})$. For every orientation measurement in vector $O$, an error is computed with Eq. (2), where $d$ is the absolute difference between the value
defined in the rules and the actual measurement. A higher absolute difference $d$ denotes a higher difference between the actual and the typical sensor orientation, resulting in a larger value of the error $e$.

$$
e = \begin{cases} 
\frac{d^4}{0.25^3}; & 0 \leq d < 0.25 \\
3d - 0.5; & 0.25 \leq d < 0.5 \\
1 & 0.5 \leq d 
\end{cases}$$

(2)

The error values form an error vector with the same size as the orientation vector. These components are summed up in order to obtain the overall error of an activity. The error values for each activity are further passed to the next layer, i.e., hierarchical aggregation.

>>Figure 2<<

3.2.1.3 Binary Classification

This method consists of as many classifiers as there are activities to be recognized. An arbitrary ML method can be used to train the classifiers. In our experiments, we empirically chose the Random Forest method as the most successful, implemented in the Weka ML tool [27]. Each classifier is trained to separate a single activity from the rest of the activities.

Each of the binary classifiers takes the feature vector described at the beginning of Section 3.2.1 as an input, and outputs the classification probability for the current activity. The probabilities of all the classifiers are merged into a vector of probabilities and are further passed to the next layer, i.e., hierarchical aggregation.

3.2.1.4 $k$NN: $K$-nearest-neighbor method

The binary classifiers rely on a correct annotation of the training data. As the data annotation was performed manually while the experiments were in progress, it can contain errors. For example, the start of a new activity is often annotated either too soon or too late, since it is hard to
determine the correct ending and starting points between two activities. In order to reduce the influence of these mistakes and to improve the overall classification accuracy, an instance-based ML method, $k$-nearest neighbor (kNN), is used as implemented in Weka [27]. kNN is instance-based (lazy) because it does not use the training data to do any generalization, but instead keeps all the training instances in the memory.

For each instance to be classified, $k$ nearest instances from the training set are retrieved. The instances labeled with each activity are then counted, forming a frequency vector. The frequency vector is normalized so that each component represents the probability information of an instance to be classified to a given activity. The resulting probability vector is passed to the next layer.

### 3.2.2 Middle-layer Aggregation

In the middle layer of the TriLAR architecture, the outputs from the bottom layer are aggregated and the activities are recognized. Figure 3 shows the proposed hierarchical structure for the middle-layer aggregation of the bottom-layer predictions. The aggregation uses three levels: meta-classification A, R-BAR and meta-classification B. The lowest two, either recognize the final activity, or recognize it as belonging to a group and pass the decision to a higher level. The structure of this layer was designed to combine domain knowledge with ML and in this way achieve better results. Domain knowledge incorporated into the R-BAR can distinguish between static activities (i.e., postures) and the Meta-classifiers A and B use the principle of multiple knowledge [28] to improve the recognition of dynamic activities.

On the first level, Meta-classifier A is trained to distinguish between cycling, transitions (such as standing up or sitting down) and other activities. All the other activities can be associated with distinct postures and can therefore be efficiently separated with R-BAR, while cycling would
sometimes be classified as sitting, bending, standing, walking or even kneeling by R-BAR. The same applies to the transitions. A feature vector of the Meta-classifier A is defined as follows:

\[ x_A = \left( \omega_{1,A} \cdot x_{kNN}^{(1)} + \omega_{2,A} \cdot x_{BC}^{(1)} + \omega_{1,A} \cdot x_{kNN}^{(2)} + \omega_{2,A} \cdot x_{BC}^{(2)} + \omega_{1,A} \cdot x_{kNN}^{(3)} + \omega_{2,A} \cdot x_{BC}^{(3)} \right) \]

(3)

where \( \omega_{1,A} \) and \( \omega_{2,A} \) (\( \omega_{1,A} + \omega_{2,A} = 1 \)) are the confidence factors in the binary and kNN classifiers’ output, determined experimentally, \( x_{kNN}^{(i)} \) is the kNN’s probability for the i-th activity (cycling, transition, other activities) and \( x_{BC}^{(i)} \) is the Binary classifier’s probability for the i-th activity. Meta-classifier A is trained using the Random Forest algorithm [29]. If a feature vector \( x_A \) is classified as cycling or transition, the aggregation is completed. If it is classified as other activities, the second level of the aggregation is applied to the bottom-layer outputs.

On the second level, the R-BAR is used to distinguish between body postures using the domain knowledge. An empirical analysis of the rules showed that they are the most suitable for distinguishing static activities (postures). The procedure described in Section 3.2.1.2 is applied to instances that were classified as other activities on the first level of the aggregation structure. Six postures are recognized on this level: lying, kneeling, sitting, bending, on all fours and upright posture. In the case of the upright posture, the third level of the aggregation procedure is used to further distinguish between the activities associated with this position; otherwise the aggregation procedure is completed.

On the last level, the instances that were classified as an upright position are further classified by Meta-classifier B into: standing, walking and running. Like with the bottom level, the components of the feature vector \( x_B \) are weighted sums of the probabilities returned by the kNN and binary classifiers for the aforementioned three activities. The main difference is that the weights (\( \omega_{1,B} \) and \( \omega_{2,B} \)) are not necessarily equal to the ones on the first level. The classifier used is the Random Forest algorithm.
When the aggregation is completed, on the first, second or third level, the recognized activity is further passed to the top layer of TriLAR for the removal of spurious transitions.

**Figure 3**

### 3.2.3 Top-layer Correction

The sequence of the activities’ output by the hierarchical aggregation is sometimes characterized by spurious transitions between different activities. This problem usually appears because the previous two layers of the AR architecture fail to take into account the continuity of the human activity. They classify each data sample in isolation and assume that there is no connection with the previous and the following data samples.

A common way to address this problem, and consequently to reduce spurious activity transitions, is to model the temporal dependence of the activities using HMMs [24]. A HMM observes the Markov property that the current system state is only dependent on the previous state of the system. The model consists of a number of hidden states and the associated transition probabilities between these hidden states. The hidden states emit events with certain emission probabilities, and these events are observed by an outside observer [30]. Our hidden states were the unknown true activities. The observed states were the activities aggregated by the middle layer. The Baum-Welch algorithm [31] is used to find the transition and emission probabilities of the HMM, and the Viterbi algorithm [32] is used to generate the most likely sequence of hidden states given an observation sequence of events. The output of the method was used to correct the aggregated middle-layer prediction and output the final decision about the user’s activity.
4 Experimental Setup

4.1 Sensor Equipment

The sensor equipment used in this study consists of three accelerometers placed on the chest, thigh and ankle. These placements were chosen as a trade-off between the intrusiveness with respect to the user and the achieved AR performance on preliminary tests [13]. For each sensor placement a custom-made body strap was used. These straps were made of elastic material with Velcro at the ends, which meant they could be adapted to different types of users. After analyzing the various available commercial accelerometers, we chose the Shimmer sensor platform [33] that has a reasonable battery life and compact size, is completely wireless, and has the option to reprogram the sensor to the user's needs and situations. The chosen platform, besides the 3-axis accelerometer, uses Bluetooth communication, and has 2GB of storage capacity, which is enough to store 3 months of sensor data, when the frequency of the data acquisition is 50 Hz. This frequency was also used to record all the experimental data described in the next section.

4.2 Experimental Data

A complex, 90-minute, test scenario was designed in cooperation with a medical expert to capture the real-life conditions of a person’s behavior, although it was recorded in a laboratory. The scenario was performed by ten volunteers. They were asked to attach the sensors to their ankle, thigh and chest, following the instructions from the expert. The scenario was divided into three groups, each containing several sub-scenarios. The first group was exercising activities. Together, the three sub-scenarios were recorded: walking on a treadmill with a one-percent inclination at 4 km/h and 6 km/h, running on a treadmill with a one-percent inclination at 8 km/h and cycling on a stationary bicycle with 65 RPM with the difficulty
set to 80 watts for the first six minutes and 160 watts for the other six minutes. In the second
group, elementary activities and transitions between the activities were recorded. The sequence
of activities performed in these sub-scenarios was predefined and volunteers were asked to
follow them. In the third group, everyday-life activities were recorded. The sequence of activities
was not predefined and the volunteers were asked to mimic their normal, everyday-life behavior
when executing activities such as cooking, reading, typing, washing dishes, scrubbing the floor,
etc.

Altogether, ten sub-scenarios were recorded, resulting in 140 recordings, as some sub-scenarios
were repeated multiple times, yielding a total of approximately 1,000,000 raw-data samples per
volunteer. These raw-data samples were transformed into approximately 7,000 feature vectors
per volunteer. The scenario included ten elementary activities (the percentage of instances per
class): standing (16%), sitting (11%), lying (22%), on all fours (10%), kneeling (6%), bending
(standing leaning) (3%), walking (15%), running (5%), cycling (10%) and transition (going
down and standing up) (2%). These activities were selected because they are the most common
elementary, everyday-life activities. A more detailed breakdown is presented in Table 1.

>>Table 1<<

4.3 Methods setup
The preliminary tests [34] showed that a 2-second window size for the sliding window is a
reasonable trade-off between the duration of the activities and the recognition delay. In some
cases, longer windows yielded higher recognition accuracies, but some short activities were
missed if the length was more than 2 seconds.

The parameters used for the SVM (implemented as SMO in Weka), Decision Trees
(implemented as J48 in Weka), Naive Bayes and Random Forest were all default ones, as
described in Weka’s API [35]. Also, the kNN (implemented as LinearNNSearch in Weka) parameters were set to default, except for the k value, which was set to 81, as proposed by Maier et al. [36]. Two other values of k (k=1, k=201) were also tested, but there was no significant difference in the final accuracy. For the middle-layer aggregation, four confidence factors were used. These factors were determined experimentally and set to $\omega_{1,A}=\omega_{1,B}=0.25$ and $\omega_{2,A}=\omega_{2,B}=0.75$.

For the top layer, the implementation of the HMM consisted of two phases. The Baum-Welch method was parameterized by the following: $N=10$ internal hidden states (equal to the number of activities due to the Viterbi assumption), the initial state-transition probability $a_{ij}=1/10$; the initial state probability $p_i=1/10$, and the output symbol distribution in state $b_j(k)=1$ if $k=j$, otherwise $b_j(k)=0$. Learning was performed with the Jahmm implementation [37] of the Baum-Welch method with 70 iterations on the training data. The size of the observation-activities sequence used by the Viterbi algorithm was experimentally set to 200.

5 Experimental Results and Discussion

To test the proposed TriLAR architecture, we used the dataset described in Subsection 4.2. The evaluation technique for the ML methods, the ones that require training a model, was the leave-one-person-out cross validation. This technique constructs the training model on the data from all the people except one. The remaining person is used to evaluate the accuracy of the trained model. This procedure was repeated for each person (10 times) and the average performance was measured. This evaluation procedure was used because using the same person’s data would give overly optimistic results if the intended use of the model is to classify the activities of previously unseen people.
Table 2 shows the AR accuracy and standard deviation achieved by the TriLAR architecture for each sensor placement. In order to show the improvements of each layer of the TriLAR architecture, the accuracies and standard deviations achieved by the methods in each layer are presented, i.e., the bottom layer − R-BAR, kNN, binary classification; the middle layer − hierarchical-aggregation; and the top layer − HMM correction. The results show that for each sensor placement, the middle and the top layers improve the basic accuracy of each of the bottom-layer methods. Additionally, the sensor-placement analysis shows that the ankle and thigh sensor placements are the best performing for one-sensor AR systems in which each activity is equally important, achieving an 85.72% and 85.96% accuracy, respectively. The chest & ankle is the best-performing sensor combination for two-sensor AR systems, which also improves the best single-sensor placement by 8.1 p.p. The three-sensor placement (chest & ankle & thigh) achieves a 98.03% accuracy, which is 3.96 p.p. better than the best two-sensor placement. However, by increasing the number of sensors, the intrusiveness with respect to the user increases as well. Therefore, the two-sensor placements can be further analyzed as a reasonable trade-off between the number of sensors and the AR performance.

In addition, the performance of the TriLAR architecture was compared to three commonly used single-layer classification methods: Support Vector Machine (SVM), Decision Tree (DT) and Naive Bayes (NB). The highest accuracies achieved by the single-layer methods for each sensor placement are marked with a grey color in Table 2. The comparison between the TriLAR final accuracy (black) and the single-layer methods (grey) shows that the TriLAR architecture outperforms the single-layer methods for any sensor placement. For instance, the TriLAR accuracy achieved by the best-performing two-sensor placement, i.e., chest & ankle, is 94.06%, which is 4.2 p.p. better than the best-performing single-layer method, i.e., SVM. In total, the
TriLAR architecture outperformed the single-layer classifiers by an average of 4 p.p. In addition, tests of the statistical significance were performed. Because of the small number of folds (10) and because the individual samples (folds) are paired (the same person's data for each placement), we used the paired Student's t-test with a significance level of 5%. The results showed that the TriLAR performance is statistically, significantly better than each of the single-layer methods.

Classification accuracy can sometimes be misleading as it averages over all the activities. This is a problem, especially when the test set is not balanced, i.e., some activities are represented by a small number of examples [38]. Therefore, in Table 3, the TriLAR recall, precision and F-measure (F1 score) for each activity and each sensor placement are presented. Because the F-measure combines the precision and the recall (harmonic mean), it is used as a reference metric for an easier discussion and explanation of the results. Thus, for each activity the statistically significant best-performing (in terms of F-measure) sensor placements are marked with black. Additionally, a set of confusion matrices is shown in Figure 4. Each confusion matrix is presented with a density plot: the darker the color, the higher the accuracy. Actual activities are placed horizontally and the predicted ones are placed vertically (the activity numbers correspond to the numbers in Table 3). The correct predictions are shown on the diagonal of each matrix. All the other positions denote misclassified examples.

The results show that the ankle and thigh sensor placements, compared to the chest, are better in almost any activity. The difference is most evident in the kneeling, sitting and cycling activities.
The rationale behind this is that the chest sensor has similar orientations and accelerations during kneeling, sitting and standing, on the one hand, and walking and cycling on a stationary bike, on the other. The confusion matrix [TriLAR × C] in Figure 4 representing the chest sensor confirms this explanation and shows the mutual misclassification between these activities (activities no. 6, 2 and 3; and 8 and 10). For the same reasons, the ankle and the thigh sensor mutually misclassify: kneeling and on all fours; and standing and bending, (shown in [TriLAR × A] and [TriLAR × T] matrices in Figure 4, activities no. 6 and 5; and 3 and 4). In two-sensor placements, the difference in the accuracies is minimal (1.2 p.p.). The best performing is the chest & ankle placement, which achieves lower accuracies only for the sitting and standing activities. The reason for this is the similarity in the sensor orientations during these two activities. This is also confirmed by the [TriLAR × A+T] confusion matrix shown in Figure 4, activities no. 2 and 3. The chest & thigh placement is better in recognizing sitting, but has a lower accuracy for kneeling and standing activities (shown in [TriLAR × T+C] matrix, activities no. 6 and 3). The reasons are the same as before, i.e., the similarity in sensor orientations. The thigh & ankle placement has a lower accuracy in kneeling and all fours for the same reasons (shown in [TriLAR × A+T] matrix, activities no. 6 and 5). To summarize, in two-sensor placements the results show that there is no sensor placement that clearly outperforms the others. Therefore, given the analysis in Table 3 and Figure 4, the most appropriate sensor placement can be selected for each use case with respect to the recognition accuracy of the individual activities and the user's comfort.

The three-sensor placement performance for different activities shown in Table 3 and Figure 4 ([TriLAR × A+T+C] matrix) shows that three sensors are enough to develop an almost 100% accurate AR system.
6 Conclusion

We have presented a novel architecture for activity recognition (AR) called TriLAR. It has a three-layer design: bottom layer (arbitrary number of independent AR classifiers), middle layer (meta-classification and domain knowledge to aggregate the predictions from the bottom-layer components), and top layer – hidden Markov model that uses the temporal dependence of the activities to remove spurious transitions between them). The architecture incorporates methods that have proven to be successful for AR and combines them in a dedicated way that each component performs the task most suitable for it.

The TriLAR architecture is a general AR concept that can be used with different activities and the data acquired from different types of sensors. We are confident that it can be used with other wearable sensors (gyroscopes, magnetometers, location sensors) and also possibly with non-wearable sensors (e.g., pressure sensors, sound, cameras, etc.). To do so, the rule-based method would require the most changes, while the ML-based bottom-layer methods (kNN and binary classifiers) would require new features. Meta classifiers in the middle layer and the top layer could remain unchanged. Furthermore, the bottom layer of the TriLAR architecture can be extended or even completely replaced with an arbitrary number of AR methods.

We tested the performance of the TriLAR architecture with all possible combinations of three accelerometer placements on a complex, 90-minute scenario. The results showed that the accuracy in the three-layer architecture improves with each layer for all the sensor placements. Additionally, the TriLAR was shown to significantly outperform three commonly used, single-layer methods: Decision tree, Naive Bayes and SVM. And finally, we showed that by using an advanced layered architecture, such as TriLAR, it is possible to successfully recognize elementary, everyday-life activities using a small number of sensors, i.e., one or two. The best
practical solution may be the ankle & chest or thigh & chest sensor placement, but the analysis of the sensor placements shown in Table 2 and Table 3 makes it possible to select the most appropriate placement for each individual case.

ACKNOWLEDGMENTS

References


FIGURE CAPTIONS

Figure 1. System overview

Figure 2. R-BAR rules for sitting, on all fours and standing

Figure 3. Middle-layer aggregation procedure

Figure 1. System overview

\[ O_{\text{sitting}} = (\frac{1}{2}, 1, \frac{1}{2}, 1, \frac{1}{2}, \frac{1}{2}) \]

\[ O_{\text{on all fours}} = (\frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}, 1, \frac{1}{2}) \]

\[ O_{\text{standing}} = (1, \frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}) \]

Figure 2. R-BAR rules for sitting, on all fours and standing
Figure 3. Middle-layer aggregation procedure
TABLE CAPTIONS

Table 1. Activity Scenario

Table 2. Classification accuracy (in percent) for all methods in TriLAR and for each sensor placement

Table 3. TriLAR classification recall or true-positive rate (in percent) for all activities for different sensor placements
Table 1. Activity Scenario

<table>
<thead>
<tr>
<th>Group of activity scenarios (percentage of instances per group)</th>
<th>Recorded activities (percentage of instances per class per group)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercising (25%)</td>
<td>walking (10%)       running (5%)      cycling (10%)</td>
</tr>
<tr>
<td>Elementary activities and transitions between them (50%)</td>
<td>lying (17%)        sitting (8%)       standing (8%)</td>
</tr>
<tr>
<td></td>
<td>bending (3%)       kneeling (6%)      on all fours (2%)</td>
</tr>
<tr>
<td></td>
<td>transition (2%)    walking (4%)</td>
</tr>
<tr>
<td>Everyday-life activities (25%), E.g. reading, typing, cooking, washing dishes, cleaning</td>
<td>lying (5%)         sitting (5%)       standing (8%)</td>
</tr>
<tr>
<td></td>
<td>on all fours (4%)  kneeling (1%)      walking (2%)</td>
</tr>
</tbody>
</table>
Table 2. Classification accuracy (in percent) for all methods in TriLAR and for each sensor placement

<table>
<thead>
<tr>
<th>Sensor placements</th>
<th>TriLAR</th>
<th>Bottom layer</th>
<th>Middle layer</th>
<th>Top layer</th>
<th>Single-layer methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R-BAR</td>
<td>kNN</td>
<td>Binary</td>
<td>Aggregation</td>
</tr>
<tr>
<td>Ankle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>76.65</td>
<td>81.14</td>
<td>83.73</td>
<td>84.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70.42</td>
<td>64.40</td>
<td>73.14</td>
<td>76.56</td>
</tr>
<tr>
<td>Thigh</td>
<td></td>
<td>82.10</td>
<td>80.59</td>
<td>83.95</td>
<td>84.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>88.86</td>
<td>83.20</td>
<td>92.49</td>
<td>92.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90.92</td>
<td>77.08</td>
<td>91.06</td>
<td>91.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>92.22</td>
<td>85.30</td>
<td>91.85</td>
<td>92.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.26</td>
<td>89.29</td>
<td>97.05</td>
<td>97.69</td>
</tr>
</tbody>
</table>
Table 3. TriLAR classification recall or true-positive rate (in percent) for all activities for different sensor placements

<table>
<thead>
<tr>
<th>Sensor body placements</th>
<th>Evaluation metrics</th>
<th>Ankle</th>
<th>Chest</th>
<th>Thigh</th>
<th>Chest &amp; Ankle</th>
<th>Chest &amp; Thigh</th>
<th>Thigh &amp; Ankle</th>
<th>Chest &amp;Thigh &amp;Ankle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Lying</td>
<td>Recall</td>
<td>94.6</td>
<td>97.4</td>
<td>92.9</td>
<td>98.3</td>
<td>99.4</td>
<td>99.6</td>
<td>99.7</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>96.1</td>
<td>96.0</td>
<td>96.1</td>
<td>98.9</td>
<td>98.6</td>
<td>99.0</td>
<td>99.1</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td>95.3</td>
<td>96.7</td>
<td>94.5</td>
<td><strong>98.6</strong></td>
<td><strong>99.0</strong></td>
<td><strong>99.3</strong></td>
<td><strong>99.4</strong></td>
</tr>
<tr>
<td>2. Sitting</td>
<td>Recall</td>
<td>80.8</td>
<td>40.7</td>
<td>81.2</td>
<td>77.8</td>
<td>97.5</td>
<td>97.6</td>
<td>98.5</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>82.6</td>
<td>57.6</td>
<td>73.7</td>
<td>83.1</td>
<td>90.9</td>
<td>96.0</td>
<td>97.7</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td>81.7</td>
<td>47.7</td>
<td>77.3</td>
<td><strong>80.4</strong></td>
<td><strong>94.1</strong></td>
<td><strong>96.8</strong></td>
<td><strong>98.1</strong></td>
</tr>
<tr>
<td>3. Standing</td>
<td>Recall</td>
<td>78.7</td>
<td>73.6</td>
<td>80.3</td>
<td>86.0</td>
<td>87.4</td>
<td>90.1</td>
<td>97.3</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>75.3</td>
<td>57.0</td>
<td>72.6</td>
<td>82.2</td>
<td>80.7</td>
<td>90.0</td>
<td>97.4</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td><strong>77.0</strong></td>
<td>64.2</td>
<td>76.3</td>
<td><strong>84.1</strong></td>
<td><strong>83.9</strong></td>
<td><strong>90.0</strong></td>
<td><strong>97.3</strong></td>
</tr>
<tr>
<td>4. Bending</td>
<td>Recall</td>
<td>56.3</td>
<td>59.5</td>
<td>66.2</td>
<td>93.2</td>
<td>84.8</td>
<td>75.3</td>
<td>96.0</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>60.4</td>
<td>61.0</td>
<td>57.4</td>
<td>95.5</td>
<td>83.1</td>
<td>73.6</td>
<td>97.2</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td>58.3</td>
<td>60.2</td>
<td><strong>61.5</strong></td>
<td><strong>94.3</strong></td>
<td>83.9</td>
<td>74.4</td>
<td>96.6</td>
</tr>
<tr>
<td>5. All fours</td>
<td>Recall</td>
<td>53.5</td>
<td>80.0</td>
<td>71.5</td>
<td>98.7</td>
<td>93.0</td>
<td>78.7</td>
<td>99.1</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>58.6</td>
<td>76.3</td>
<td>74.7</td>
<td>94.7</td>
<td>94.9</td>
<td>81.6</td>
<td>98.7</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td>55.9</td>
<td><strong>88.1</strong></td>
<td>73.1</td>
<td><strong>96.7</strong></td>
<td><strong>93.9</strong></td>
<td><strong>80.1</strong></td>
<td><strong>98.9</strong></td>
</tr>
<tr>
<td>6. Kneeling</td>
<td>Recall</td>
<td>63.2</td>
<td>31.2</td>
<td>46.1</td>
<td>98.8</td>
<td>63.0</td>
<td>84.5</td>
<td>98.8</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>54.1</td>
<td>36.7</td>
<td>54.5</td>
<td>99.1</td>
<td>79.8</td>
<td>81.0</td>
<td>99.1</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td><strong>58.3</strong></td>
<td>33.7</td>
<td><strong>49.9</strong></td>
<td><strong>98.9</strong></td>
<td>70.4</td>
<td>82.7</td>
<td><strong>98.9</strong></td>
</tr>
<tr>
<td>7. Transition</td>
<td>Recall</td>
<td>66.4</td>
<td>77.8</td>
<td>65.5</td>
<td>75.2</td>
<td>69.2</td>
<td>68.2</td>
<td>80.6</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>72.8</td>
<td>77.2</td>
<td>70.7</td>
<td>77.1</td>
<td>72.9</td>
<td>75.5</td>
<td>84.7</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td><strong>69.5</strong></td>
<td><strong>77.5</strong></td>
<td>68.0</td>
<td><strong>76.1</strong></td>
<td><strong>71.0</strong></td>
<td><strong>71.7</strong></td>
<td><strong>82.6</strong></td>
</tr>
<tr>
<td>8. Walking</td>
<td>Recall</td>
<td>97.2</td>
<td>95.2</td>
<td>97.4</td>
<td>97.3</td>
<td>97.3</td>
<td>97.7</td>
<td>98.3</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>96.2</td>
<td>93.4</td>
<td>95.8</td>
<td>96.3</td>
<td>96.3</td>
<td>97.1</td>
<td>97.4</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td><strong>96.7</strong></td>
<td><strong>94.3</strong></td>
<td><strong>96.6</strong></td>
<td><strong>96.8</strong></td>
<td><strong>96.8</strong></td>
<td><strong>97.4</strong></td>
<td><strong>97.8</strong></td>
</tr>
<tr>
<td>9. Running</td>
<td>Recall</td>
<td>96.8</td>
<td>97.3</td>
<td>97.3</td>
<td>97.1</td>
<td>97.1</td>
<td>97.1</td>
<td>97.6</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>98.9</td>
<td>98.9</td>
<td>99.9</td>
<td>98.6</td>
<td>99.6</td>
<td>99.8</td>
<td>99.8</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td><strong>97.8</strong></td>
<td><strong>98.1</strong></td>
<td><strong>98.6</strong></td>
<td><strong>97.8</strong></td>
<td><strong>98.3</strong></td>
<td><strong>98.4</strong></td>
<td><strong>98.7</strong></td>
</tr>
<tr>
<td>10. Cycling</td>
<td>Recall</td>
<td>99.8</td>
<td>83.9</td>
<td>99.1</td>
<td>99.8</td>
<td>99.7</td>
<td>99.8</td>
<td>99.9</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>99.9</td>
<td>86.4</td>
<td>99.7</td>
<td>100.0</td>
<td>99.7</td>
<td>99.9</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>F-measure</td>
<td><strong>99.8</strong></td>
<td>85.1</td>
<td><strong>99.4</strong></td>
<td><strong>99.9</strong></td>
<td><strong>99.7</strong></td>
<td><strong>99.8</strong></td>
<td><strong>99.9</strong></td>
</tr>
</tbody>
</table>