IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 March 2024, accepted 1 April 2024, date of publication 8 April 2024, date of current version 17 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3386351

RESEARCH ARTICLE

Adaptive Hierarchical Classification for Human Activity Recognition Using Inertial Measurement Unit (IMU) Time-Series Data

HEBA NEMATALLAH, (Graduate Student Member, IEEE), AND SREERAMAN RAJAN, (Senior Member, IEEE)
Systems and Computer Engineering Department, Carleton University, Ottawa, ON K1S 5B6, Canada

Corresponding author: Heba NematAllah (hebanematallah@cmail.carleton.ca)

This work was supported in part by Ontario Graduate Scholarship Program and the Discovery Grant Program of Natural Science and Engineering Research Council (NSERC) of Canada.

ABSTRACT Human Activity Recognition (HAR) based on Inertial Measurement Unit (IMU) has become increasingly important in health and fitness applications. These systems can continuously and cost-effectively monitor human activity, regardless of the surrounding environment. However, the current dominant trend in HAR uses black box flat classification (FC) methods, which lack interpretability and do not consider the natural hierarchical relationship between activity classes. Such systems often achieve greater accuracy at the cost of increased complexity and are not suitable for critical decision-making applications. This paper proposes an Adaptive Hierarchical Decision Tree (AHDT) HAR system that recognizes human activities based on IMU measurements along with quasi-stationary inclination feature extraction. The proposed system generates a global classifier that classifies human activities according to a tree taxonomy structure. This approach maintains interpretability while considering the fundamental signal data features embedded in the natural hierarchical representation of the activity classes. In addition to the commonly used flat classification accuracy measures, we applied modified hierarchical accuracy measures to assess the exclusive characteristics of hierarchical relationships between the classes. We used seven publicly available datasets to evaluate the proposed system and compared its performance with other tree-based classifiers, including Random Forest, Gradient Boosting, XGBoost, and AdaBoost classifiers. Our results demonstrated that the AHDT system significantly improves the recognition performance of fine-grained activities and offers a balance between lower complexity and higher interpretability. Overall, the proposed AHDT system provides an interpretable and practical approach to HAR that can be valuable in critical decision-making applications.

INDEX TERMS Digital signal processing (DSP), decision tree (DT), flat classification (FC), human activity recognition (HAR), hierarchical classification, hierarchical f-measure (HF), hierarchical precision (HP), hierarchical recall (HR), inertial measurement unit (IMU), interpretable machine learning.

I. INTRODUCTION

Health applications related to ageing are critical, as in July 2021, Canada's senior population, 65 and older, reached about 18.5% of the total population [1]. This percentage is expected to grow by 68% by 2037 [2]. This growth imposes severe stress on the healthcare system to provide professional elderly care due to limited financial and human resources. This demand for monitoring the ageing population has resulted in increased active research in the area of human activity recognition (HAR). HAR systems provide healthcare providers with information on patients' activity

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

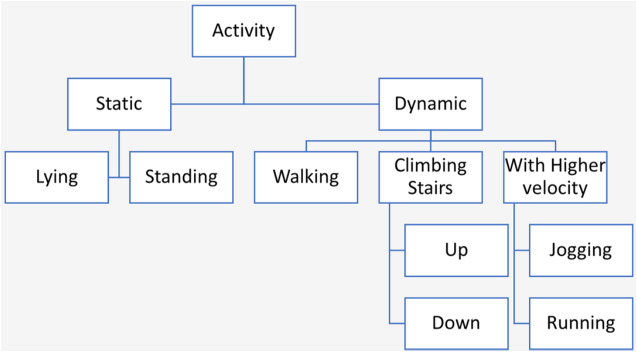


FIGURE 1. An example of tree/ hierarchical taxonomy of human activities.

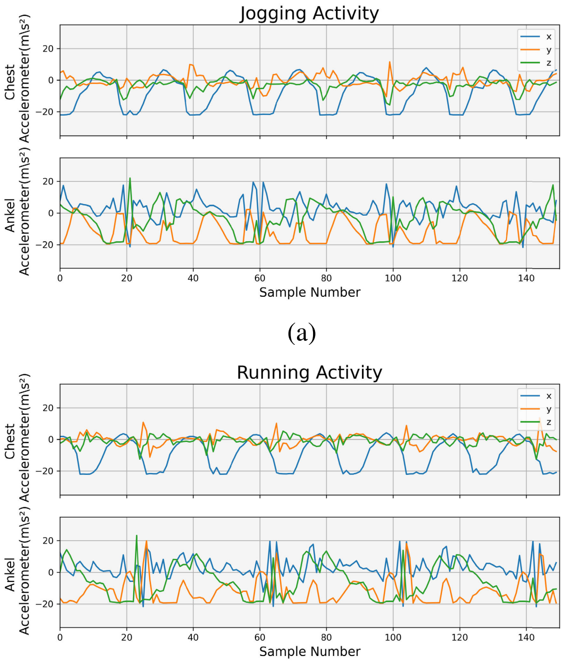
and activity levels. HAR systems aim to detect human body motion and activity patterns based on wearable sensor data. These systems form the foundation for many real-life applications, such as healthcare monitoring [3], assistive living [4], and activity awareness [5]. Such systems integrate machine learning and digital signal processing for activity sensing and recognition.

There are several factors to consider when selecting the most suitable sensing technique for monitoring human activities. These factors include cost, computational complexity, recognition rates, and privacy preservation. Wearable devices such as smartphones, smartwatches, wristbands, and smart glasses are often used for Human Activity Recognition (HAR) to monitor human activities continuously [6]. These devices have different built-in sensors, including accelerometers and gyroscopes, which comprise the Inertial Measurement Unit (IMU) [7].IMU-based HAR systems have the advantage of being less expensive than other sensing devices. Additionally, they are non-intrusive and do not compromise users' privacy or interact with their surroundings.

Various human activities exhibit similar sensor patterns, making it difficult to distinguish between them using conventional flat classification methods [8]. Flat classification refers to a classifier trained on all categories simultaneously and at the same level. The complicated nature of the sensor patterns poses a significant challenge for such methods, increasing the need for techniques to accurately classify the different types of human activities with less complicated complexity.

Human activities can be divided into three main categories: static, low dynamic, and dynamic activities. Static activity involves minimal movement or physical effort, such as being steady in the environment. Standing still, sitting, and lying are examples of static activities. In contrast, an activity is considered low dynamics if there is minimal physical movement, but it is not wholly inactive. Simple household chores like sweeping and vacuuming are examples of low-dynamic activities. In comparison, the dynamic activity involves movement, often in a constantly changing environment, such as running and performing sports.

A hierarchical structure with interclass distinguishability exists while designing a human activity classification system.



(a)

(b)

FIGURE 2. Examples of accelerometer data for two different activities (a) Jogging, (b) Running.

Ignoring such a structure while classifying different activities can result in a classifier struggling to differentiate between activities with similar patterns, leading to misclassification. The examples shown in Figure 2 demonstrate this problem, where running and jogging have similar sensor patterns, making it difficult for a single classifier to identify both classes correctly. On the other hand, climbing stairs up is a dynamic walking activity with an upward inclination, while jogging is a dynamic walking activity without inclination but with a higher pace. Additionally, standing and laying are considered static activities that can be better represented using a hierarchical taxonomy, as illustrated in Figure 1. By considering the hierarchical nature of human activities, we can develop more accurate and efficient classifiers that recognize a broader range of activities.

Hierarchical classification (HC) is a multi-class classification decomposition approach that splits the multi-class problem into a set of condensed classification problems utilizing the shared characteristics and inherited relations among the classes. The classes in the HC are grouped according to a hierarchy(tree) or -in rare cases- directed acyclic graph (DAG) [8]. Although class-hierarchy representation seems naturalistic, there is only minimal research on representing those relationships by machine learning algorithms. This inadequacy in research becomes very obvious when comparing the number of papers published about flat HAR classification to hierarchical classification approaches [3], [4], [5], [6], [7], [9], [10].

A modified decision tree algorithm to support hierarchical classification was proposed earlier in the literature, but it had limited efficiency, especially when labelled data is scarce. In [11], the HMC4.5 hierarchical method was presented as a variation of the C4.5 tree induction algorithm with modified entropy calculations. In the HMC4.5 method, the reformulation of the entropy considers the information of the descendant classes of a given class in the hierarchy while incorporating the tree size in the entropy formula. This process of tree induction is regarded as a complex task [8], and it is acknowledged that the discrimination of all classes simultaneously can lead to suboptimal results [8]. Also, it can result in large tree sizes and increased computational complexity. As a result, better modification of the DT-based HC algorithm than HMC4.5 is needed.

This research proposes an Adaptive Hierarchical Decision Tree (AHDT) HAR system, where the predicted human activities are organized into a tree taxonomy structure. This proposed method manages to preserve natural information about the time signal data hierarchy while maintaining the practicality of design and lower complexity. The proposed hierarchical classifier helps distinguish between the various activities that can cause ambiguity in classification by giving a more profound comprehensive activity recognition without sacrificing accuracy. In the proposed method, we use adaptive segmentation to adjust the segmentation parameters based on the dynamics of the activity's time series data, resulting in more accurate detection of different movements. With adaptive segmentation, we can automatically identify periods of static activities, low dynamic or dynamic ones, which can help increase the model's predictive ability.

Further, the proposed hierarchical classifier differs from the state-of-the-art hierarchical classifiers, where a quiet number of ensemble classifiers are combined to obtain the global classification. The proposed classifier detects the activity category using one classification model with different levels of abstraction, which minimizes the system's computational complexity while maximizing its generalization capabilities simultaneously. In addition, the proposed Hierarchical system overcomes the limitations of the earlier variation of decision tree-based hierarchical classification proposed in the literature.

The main contributions of this paper are summarized as follows:

1) A novel Adaptive Decision Tree-based global hierarchical classifier for IMU-based HAR systems is introduced in this work. This global hierarchical classification framework offers activity classification with varying detail levels, allowing for easy customization and generalization across a range of real-life applications.
2) A modified version of the Percentage Coverage (PC) algorithm is introduced to consolidate the local classifiers into a global classifier, resulting in minimum machine learning modifications while eliminating possible overfitting.
3) An adaptive data segment length and feature set based on activity dynamics, along with the inclusion of an IMU-based quasi-stationary inclination feature, are proposed to improve activity recognition accuracy.
4) Theoretical and experimental analyses of the computational complexity of the proposed AHDT system for classification tasks are provided.
5) Exhaustive experimental results using seven publicly available datasets, namely WISDM, HARTH, MHEALTH, DaLiAc, PAMAP2, HAR70+, and REALDISP, to demonstrate the classification model's generalization capabilities are presented.
6) Evaluation of the experimental results, using commonly used flat accuracy measures and hierarchy-based performance metrics like hierarchical precision (HP), hierarchical recall (HR), and hierarchical f-measure (HF) to account for the hidden errors in the hierarchical model, is performed.

The remainder of this paper is organized as follows. In the next section, related works and state-of-the-art are discussed. The proposed system and framework, including the classifier design, signal processing, and feature extraction, are described in Section III. Experiments and the result analysis are given in Section IV. Finally, the conclusion and future research directions are provided in Section V.

II. STATE OF THE ART

There are mainly three phases of any HAR Classification system. The first phase of any HAR approach is time signal data acquisition and segmentation, regardless of which classification method is used. The second phase, feature extraction and selection, and the third phase, Classification model building, distinguishes between the different classification approaches. After acquiring the input signal from the sensor, the time data could be segmented using fixed or variable-size windows. In the variable size data segmentation, the window size is determined based on the discriminative pattern of each activity, making it a challenge to implement for real-life systems because of the lack of prior knowledge of the movement type ahead of time [12]. Some researchers use the Dynamic Time Warping (DTW) method for variable segmentation [13] while others use an unsupervised offline approach based on deep learning for segmentation [14].

On the other hand, the window size for fixed segmentation may be selected after empirical trials [12]. If the window size is not large enough to capture information to generalize the discrimination of the activity, it will increase the probability of misclassification. A larger window may contain data for more than one activity, reducing the system's ability to detect short-duration motions, e.g., standing at the transition from sitting to walking. Researchers use overlapping segments to minimize the effect of the approximate segment length. The data segmentation into overlapping fixed-size windows has shown better accuracy than other segmentation methods and is widely used in most HAR-related research [15]. The popularly used data segment length ranges from 3 -7 sec to be large enough to capture the various activity patterns [12].

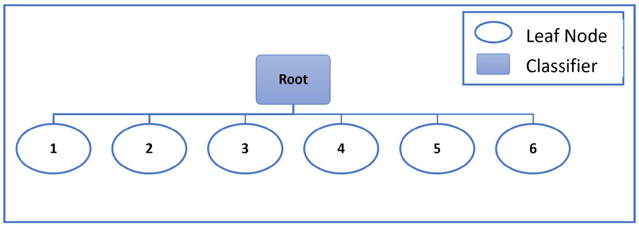


FIGURE 3. Tree/ hierarchical taxonomy used for flat classification (FC).

A. FLAT CLASSIFICATION-BASED HAR SYSTEMS

In flat classification (FC), each instance from the training dataset is associated with one class within a set of activity classes arranged at a flat level without considering any inherited parent-child relationship, as shown in Fig. 3. Most related works on HAR classification utilize the flat classification approach. Although the flat classification methods require minor adaptation to apply to different application domains, their performance decreases with increasing the number of classification categories [8], especially for classes with higher correlation (closer hierarchical relation), e.g., walking, jogging, and climbing stairs. Such models neglect essential signal data features embedded in the natural hierarchical representation of the activity classes.

In the following sub-subsections, we will discuss common flat approaches used in HAR according to the classification algorithm. An illustration of common flat classification frameworks is shown in Fig 4, which includes Conventional Classification, Deep learning-based classification, and integrated classification. A brief explanation of each framework is given in the next subsections.

1) CONVENTIONAL CLASSIFICATION

The first phase of the HAR system, namely the Data Segmentation phase, is common to conventional HAR and Dl-based HAR systems. The second phase of HAR systems, feature extraction and selection, differentiates conventional HAR systems from deep learning (DL) based ones. This phase aims to assemble a discriminative feature vector for each activity pattern. The extracted features could be time domain statistical features, frequency domain features, or both or any other domain features. This step is followed by selecting the relevant features which significantly impact system accuracy and generalization ability. Efficient feature extraction and selection algorithms will reduce processing time and increase accuracy with higher generalization to new data.

Conventional HAR systems require the features to be handcrafted or engineered. Efficient feature engineering involves extracting features for modelling the entire data segment, called global features, and others for modelling the changes within the time segment, called local features. Feature extraction usually is followed by feature selection methods to identify the most relevant features. The idea behind using the feature selection method is to increase the classification systems' performance by reducing the features' dimensionality. Measuring the features' importance/relativity and choosing the optimal feature selection method is complex as it is done using systematic experimentation. This phase also requires domain knowledge and expertise. Several classification algorithms are utilized in the literature for HAR systems. In [16], the authors used a Support Vector Machine (SVM). At the same time, in [17], a two-stage Hidden Markov Model (HMM) showed a comparable accuracy to SVM using the same dataset and a time series approach. The logistic model tree (LMT) showed good performance and generalization accuracy when used for activity detection [18].

Although conventional HAR systems showed reasonable accuracy, domain expertise and excellent knowledge of signal processing techniques are required to get the best feature from the sensor time series data. The effect of the feature extraction phase on the activity recognition process has been well-researched in the literature [19]. In addition, classification accuracy deteriorates when trying to predict fine-grained activities with the features suggested in the literature.

2) DEEP LEARNING-BASED CLASSIFICATION

The Deep Learning (DL)-based HAR systems automatically learn features from training data. The main challenging requirement of these systems is to obtain a much higher significant amount of training data than required for training the conventional HAR systems [20]. Also, hyper-parameter optimization, such as selecting the number of hidden layers, can be a time-consuming, challenging task that depends on trial and error and is usually associated with obtaining better generalization. Despite this, DL-based HAR systems have gained popularity because they can identify more complex activities without the dilemma of choosing the best feature extraction and selection methods.

Deep learning methods for classification, such as Deep Recurrent Networks (DRN) [21] and Convolutional Neural Networks (CNN), have been widely proposed. The CNN-based systems research focused on selecting the representation of the input signal or the architecture of the CNN itself [22], [23], [24], [25]. The authors in [26] and [27] represented multiple sensor inputs as one data image.

On the other hand, some researchers are interested in modifying and adapting CNN architectures. In [26], the authors used one convolutional and two fully connected layers. In [26] and [27], two convolutional and one fully connected layers are used. In [28] and [29], the authors have applied deeper architectures, undesirably leading to data overfitting. Reference [29] showed that increasing the number of hidden layers of the CNN does not always lead to higher accuracy. The choice of the number of layers or the type of deep network cannot be justified in many instances.

Deep learning models can perform complex tasks on sensors-based data but are difficult to interpret. The complexity and size of deep learning models make it difficult to determine what the models learn and how they infer their

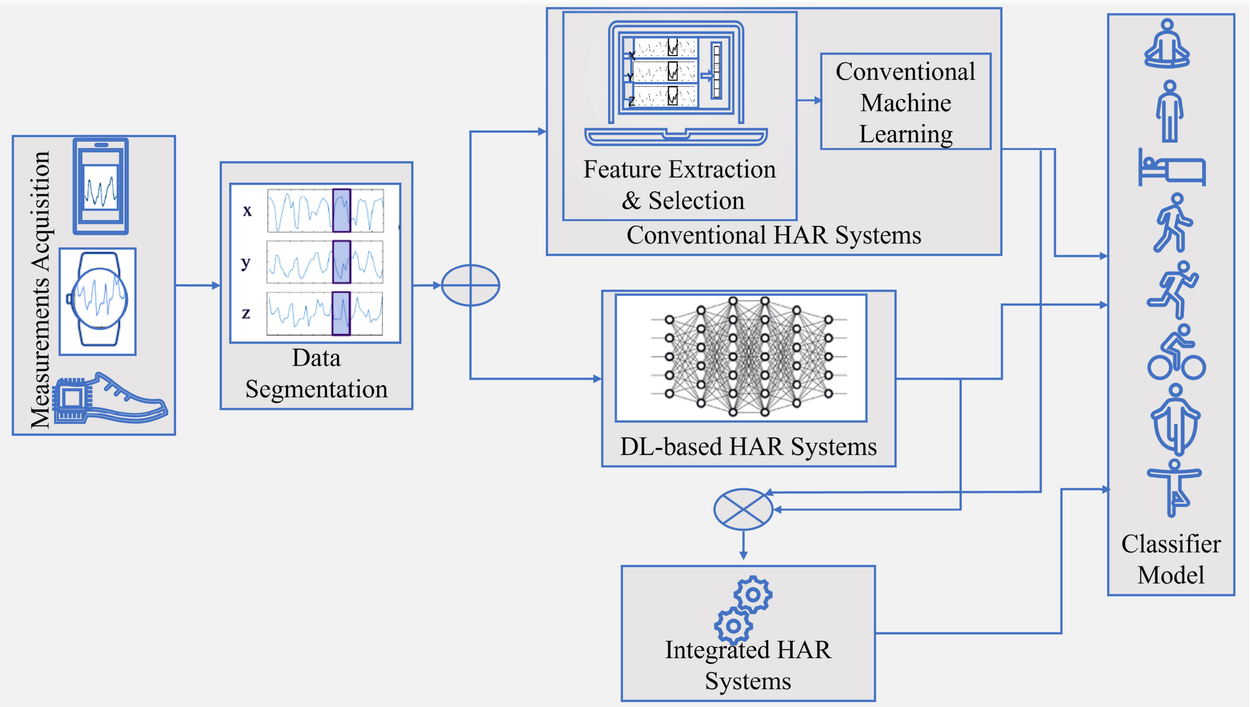


FIGURE 4. Various HAR flat classification (FC) frameworks.

classification results. It is hard to explain how the features are used to reach the prediction of classes because of the nonlinear transformations added to the input data at each layer. These systems thus turn out to be black boxes and provide little confidence to users in terms of accountability when used for critical application domains. Further, when these complex models make classification errors, it becomes difficult to ascertain whether the error was inherently due to the model itself or due to the mislabelling of data in the training set [30]. Thus, there is an undeniable need for choosing an interpretable classification algorithm for critical application domains, especially when the lives of humans are involved.

3) INTEGRATED CLASSIFICATION

Integrated classifiers are built by stacking more than one classification algorithm without considering any relation between the classification classes. Each component classifier of the integrated classifier usually becomes responsible for increasing the accuracy of one or more phases of the classification process. This makes such integrated classifiers also considered a type of flat classification, and naming it hierarchical classification in some research papers is not entirely precise.

The hierarchical classification term is misused in many papers, making it essential to distinguish between hierarchical classification and integrated classification. Hierarchical classification is where the classification method is defined over a domain-based predefined class taxonomy, which will be described in detail in the following subsection. Some existing work like [17], [21], [31], and [32] are presented as hierarchical classification. However, existing works ignore the class taxonomy, which makes them flat classifiers using multiple integrated classifiers. For example, [17] used Two-Stage Continuous HMM while [31] used Principal Component Analysis (PCA) integrated with SVM, whereas [32] used Stacked Autoencoders with SVM for activity classification. Moreover, others applied different hierarchical structures of deep learning methods, ignoring the highly elevated complexity of classifiers. For example, [21] used the Bidirectional Long Short-Term Memory (Bi-LSTM)-Based DRNN Model. Such integrated classification models can significantly increase the complexity of the recognition system. Also, they often require more training data to operate effectively. Furthermore, due to their high processing requirements, such systems are generally unsuitable for real-time applications. Although the accuracy measure does not sufficiently represent the classification algorithm performance, we have listed the accuracy measures of additional research on some publicly available datasets in Table 1.

B. HIERARCHICAL CLASSIFICATION (HC)

Many significant real-world classification problems are naturally modelled as hierarchical classification problems, where the classes to be predicted are organized into a tree class hierarchy. Minimal research exists in the hierarchical classification of HAR compared to the research done in other areas like text categorization [48] and gene prediction [49].

Few of these HAR-related hierarchical research considered automatic tree class taxonomy construction instead of knowledge/user-based methods [8]. In the automatic

TABLE 1. Survey on some of the previous research studied HAR publically available datasets.

Reference	Dataset	Method	Accuracy (%)
[33]	DaLiAc	Modified Feature extraction + Gradient Boosting Decision Tree (DT)	93
[34]	DaLiAc	CNN	95.7
[35]	DaLiAc	Inertial sensor signal to image encoding + CNN	96.4
[36]	DaLiAc	Handcrafted features + deep convolutional neural networks (DCNNs)	97.2
[37]	DaLiAc	Logistic Regression (LR)	97.3
[38]	PAMAP2	CNN	91.4
[39]	PAMAP2	MLP+CNN + LSTM	93.52
[40]	PAMAP2 WISDM	CNN +temporal attention submodules	93.16 98.85
[41]	MHEALTH WISDM	CNN+LSTM	94.86 90.41
[42]	PAMAP2 MHEALTH	Multi-head attention Graph neural networks (MhaGNN)	96.74 98.65
[43]	HARTH	Multilayer Perceptron (MLP)	92.92
[44]	REALDISP	Quaternion-based filter+orthogonal fuzzy neighborhood discriminant analysis technique+LSTM	87.35
[45]	HAR70+	Relief feature selection+ SVM	87.2
[46]	WISDM PAMAP2	Optimized 1D-CNN	97.8 90.27
[47]	WISDM	Multilevel CNN+ Wavelet Transformations	99.35

construction, Machine learning methods were used to detect class similarity. In [50], a clustering method was used to construct the class hierarchy automatically, followed by training Bi-directional CNN from data segments of 2.56 sec. However, the resulting class hierarchy doesn't represent the parent-node structure. Instead, it reflects the class similarity grouped in levels (clusters). In [9] used the normalized confusion matrix of LSTM in applying a hierarchical agglomerative clustering algorithm for automatic tree taxonomy building without providing enough details. The authors in [51] showed that using any HC produces better performance than FC. It also showed by examining different hierarchy taxonomies that changing the taxonomy will not yield significant improvement.

The test phase of all HC methods usually follows a top-down process to classify new samples in a hierarchical manner, level by level. Researchers often use standard flat classification evaluation measures instead of hierarchical ones in the literature. This adds to the limitation of the research in the area of hierarchical HAR because the flat accuracy measures cannot inspect misclassification errors at the different levels of the classification hierarchy.

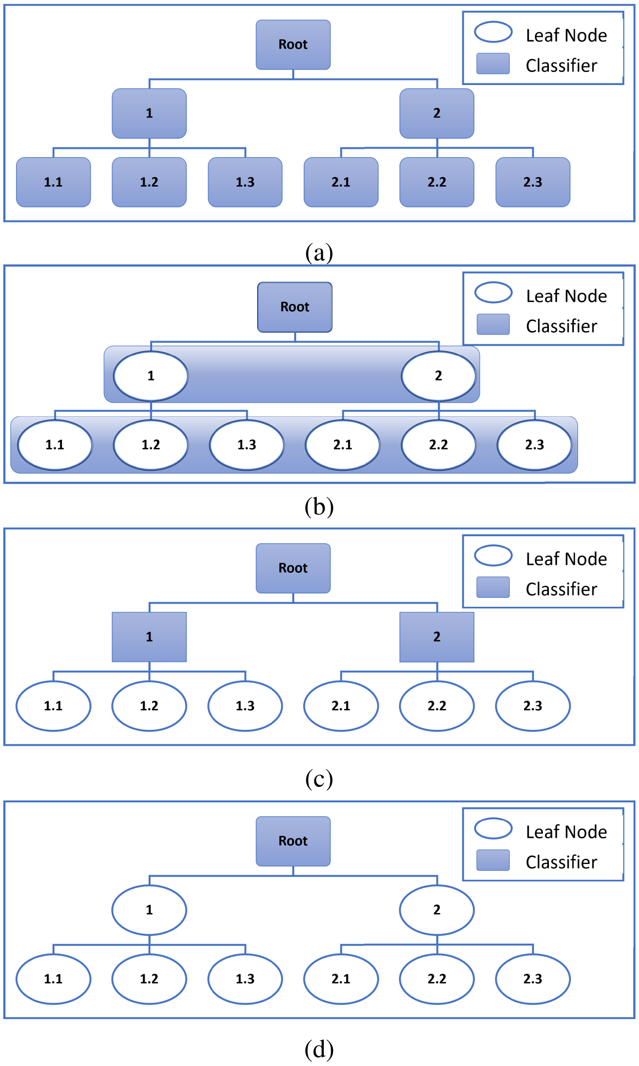
The distinction between the different hierarchical classification approaches lies in the way the hierarchical classifier is built. The building of hierarchical classifiers could be divided into the Local Classifier and Global Classifier approaches. Fig. 5 shows the different HC approaches. Each approach is explained in the following subsections.

1) LOCAL CLASSIFIER APPROACH

This approach uses the hierarchy information in the data while retaining simplicity and generality by integrating multiple local classifiers [16]. It divides the classification problem into a set of flat classification problems grouped in a tree of classifiers according to the domain taxonomy. The local flat component classifiers are arranged and trained to produce a hierarchy of predictions. The arrangement of the local classifiers could follow Local Classifier per Node (LCN), Local Classifier per Level (LCL) or Local Classifier per Parent Node (LCPN).

In the Local Classifier per Node (LCN) approach, a binary classifier is trained per each node. This approach could achieve higher accuracy than any other one because the component classifiers are more inclusive, possibly leading to higher classification accuracy. However, it suffers from high complexity due to the high number of required component classifiers. In [52], the authors used multiple binary deep Neural networks (DNN) following the LCN approach. They used one binary DNN classifier for each class with additional classifiers for each new parent class. They start classifying large segments of signal (30 sec), then splitting into smaller segments (5 -15 sec) to fine-tune the label assignment. However, experimental results show that the proposed method improves the F1 score by 2% compared to fixed windowing segmentation and flat classification. Unfortunately, this hierarchical classification model requires exceptionally high memory and has a very high computational complexity.

Another configuration is Local Classifier per Level (LCL), which requires a smaller number of classifiers; the number of classifiers equals the depth of the hierarchy tree. Still, these classifiers cannot distinguish between activities that lead to ambiguous classification. The literature on the other applications of ML shows that LCN could produce better performance than flat classification [8]. Still, it also introduces higher time and memory complexity for the increasing number of classification models involved. An example of LCL in a slightly different field is the system proposed by [53], which is a two-stage fuzzy inference system to classify the driving style using IMU data integrated with GPS data and vehicle-related data. The first stage recognizes the context where the driver is moving, and the second classifies driver behaviour as normal, aggressive or very aggressive.



(a)

(b)

(c)

(d)

FIGURE 5. Hierarchical Classification (HC) approaches (a) Local Classifier per Node (LCN), (b) Local Classifier per Level (LCL), (c) Local Classifier per Parent Node (LCPN) and (d) Global Classifier (GC).

The Local Classifier per Parent Node (LCPN) provides the trade-off between the performance and the complexity of the other two types of local hierarchical classification. A classifier for each parent node would enable the model to find the best feature set representing the fine separations between similar classes. For example, the difference between jogging and running activities is the pace (speed). When flat or LCL classification methods are used, such types always suffer from high misclassification rates. Although LCPN could propagate the errors from one level to the next, its overall accuracy would still be higher than the flat classification [51]. This approach is preferable for researchers in the HAR domain who consider hierarchical classification for its lower complexity [13].

The work in [54] used a two-level hierarchy with six labels (lying down, sitting, standing in place, walking, running, and bicycling), divided into stationary and nonstationary activity classes. The two-level hierarchy in this work refers to the taxonomy (tree) being two levels and not the structure of the classification algorithm. The testing and training are done on a publicly available dataset named the Extrasensory dataset. The Extrasensory dataset is a dataset that has data from a large number of sensors: accelerometer, gyroscope, magnetometer, watch compass, location and audio. They followed the LCPN approach by using three deep neural networks (DNN) for each parent node. The achieved balanced accuracy was 85.2 %. An LCPN approach was utilized in [55] for classifying only six classes under three parent nodes. For each local classifier, different feature subsets were selected using a fast correlation-based filter (FCBF) and classified by separated naïve Bayes classifier methods. The final leaf-predicted class label was chosen according to the maximal probability rule of the three classifiers' results. The reported accuracy using flat classification measures on a publicly available dataset, PAMAP2, was 97.96%. In this LPCN method, testing any instance required going through three classifiers with three different feature sets and then, finally, the probability rules. This resulted in high computational complexity. In [51], they empirically compared multi-class decomposition methods, which are one-vs-one, one-vs-all and hierarchical classification. The results showed the multilevel hierarchical classification outperformed the other multi-class decomposition methods.

2) GLOBAL CLASSIFIER APPROACH

In the Global Classifier approach, a single classifier handles all the specificities of an HC, usually using heavily adapted ML algorithms to consider a hierarchy of classes. These global classifiers are problem-specific classification methods that usually suffer from high computational complexity [8]. To our knowledge, none of the HAR research has used the Global Classifier approach.

III. PROPOSED METHOD

The proposed method is based on the decision tree algorithm, a predictive model that generates a series of rules to induce the predicted classification output. In health and quick response applications, where decisions are made based on the outcome of the HAR system, it is valuable to obtain more interpretable classification models like decision trees. The accumulated learning during the decision tree training is represented in the form of rules that are easy to understand and interpret. We are using the decision tree classification algorithm as the base algorithm to generate a classification model that can reveal the training data's internal information and significant attributes. This information is generated as *if-then* rules for classifying any new unseen data. So, the decision tree algorithm generates an easy-to-understand tree structure and classification rules. We use these classification rules during testing/validation or even the real-time classification phase instead of running complex classification models like deep learning models that require massive computational complexity.

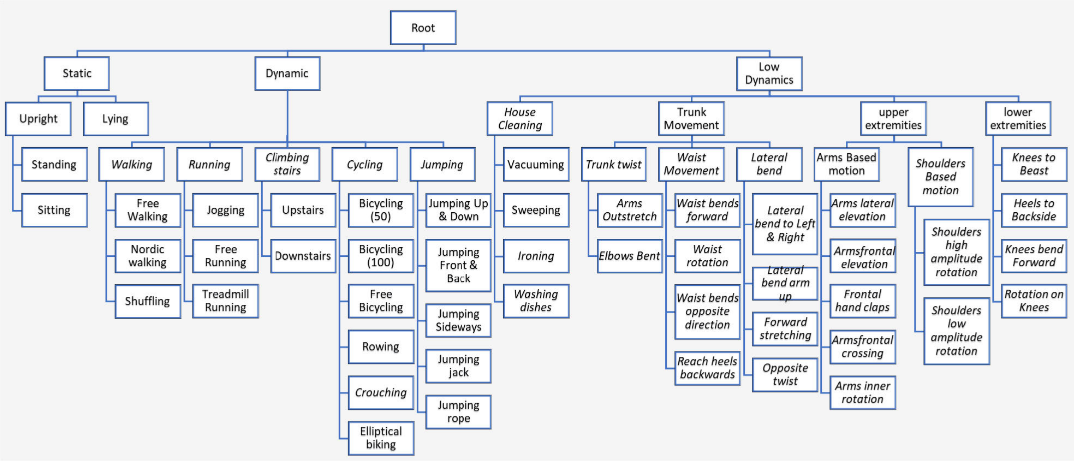


FIGURE 6. Full tree taxonomy for the proposed AHDT system.

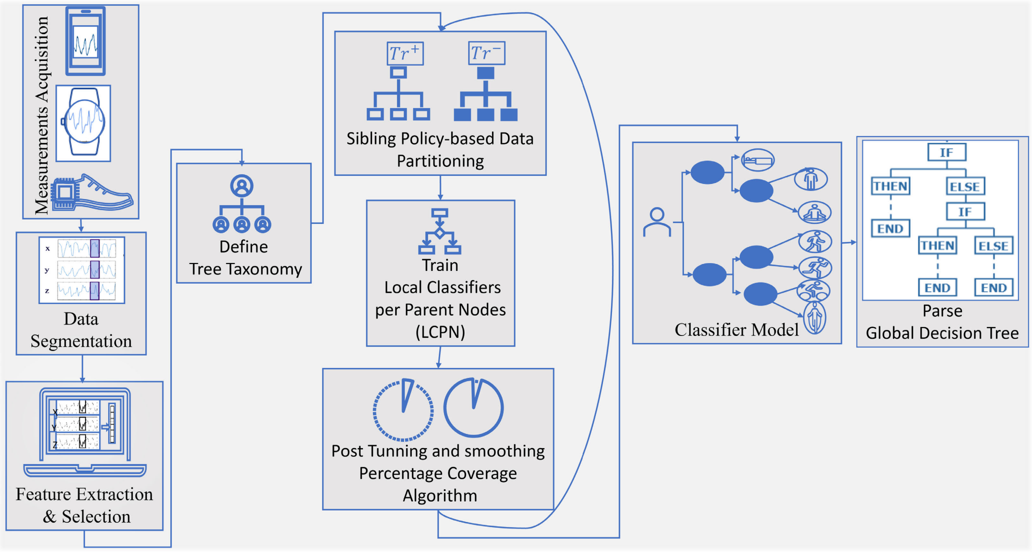

Measurements Acquisition
x
y
z
Data Segmentation
Feature Extraction & Selection
Define Tree Taxonomy
Tr^+
Tr^-
Sibling Policy-based Data Partitioning
Train Local Classifiers per Parent Nodes (LCPN)
Post Tunning and smoothing Percentage Coverage Algorithm
Classifier Model
IF
THEN
ELSE
IF
END
THEN
ELSE
END
END
Parse Global Decision Tree

FIGURE 7. Block diagram of the training phase of the proposed classifier.

A. DECISION TREE BASED HIERARCHICAL CLASSIFICATION

We adapted a Decision Tree-based Hierarchical Classification (DTHC) approach. In the hierarchical approach, each instance would belong to multiple classes organized in a hierarchy branch for continuous precise classification. In our HC, the classes are organized in a hierarchy or a tree, as shown in Fig. 6. Classifying any instance would result in concurrently belonging to multiple classes. Hence, an example that belongs to any class automatically belongs to all its parent classes. So, at any time, the instance will be assigned to different abstraction levels, which results in knowing multi-levels of information about the sample instead of total misclassification results. Fig. 7 shows the block diagram of the proposed AHDT system's training phase.

The CART machine learning algorithm has been used in this work as the base classification method for all of the local classifiers used in our initial LCPN hierarchical classifier, which is tuned and smoothed to construct the global classifier. The CART Decision Tree Algorithm is one of the most efficient, powerful, and popular Decision Tree Induction algorithms. Ross Quinlan developed this Classification Algorithm to extend his ID3 Algorithm, using the information gain ratio to select the best attribute [56]. This method's classification tree follows a top-down strategy based on the divide-and-conquer approach. The splitting measure of the tree is the Information Gain Ratio (IGR) of each feature. Starting from the root of the decision tree, the child node is generated to represent a specific threshold value of a particular data feature that maximizes the pure splitting of the classes, which is the one with the highest IGR. The tree splitting continues until all the training samples at the current node belong to one class or if there are no more data features to split. As a result, each path from the root to the leaf node formulates a decision rule.

Accordingly, at each node n, the algorithm searches for a threshold value for each of the time-series signal features/attributes A_j. This threshold called the Candidate Cut Point (CCP), should maximize the information gained from splitting the current training set (S) into two subsets (S_1, S_2) based on this CCP.

Once the threshold value is selected, the algorithm calculates the gain ratio to compare the discriminative ability of the candidate attributes and determines the optimal one to split the current node where the gain of the attribute is calculated using equation 1

$$Gain\,(A) = Entropy\,(S) - Entropy_A\,(S) \tag{1}$$

where

$$Entropy\,(S) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{2}$$

$$Entropy_A\,(S) = \sum_{j=1}^{n} p_j\; Entropy(S_j) \tag{3}$$

$$P_i = \frac{|S_i|}{|S|} \tag{4}$$

where A refers to the attribute, S is the dataset, S_j is the partion of dataset S that belongs to class j, n is the number of partitions, m is the total number of classes and $|S|$ is the number of samples in the data set S. The attribute that maximizes the information is selected as the best splitting feature, and the algorithm splits the current node by the attribute chosen threshold.

After building the classification model of each parent node, we merge these decision trees using the Percentage Coverage (PC) Post Pruning method, which will be discussed in the following subsection. The resulting classification model of the hierarchical decision tree is one decision tree that can classify new activities in a branch of the hierarchy at once. For example, a testing instance could be classified as a non-stationary activity of type walk.

B. DATA SEGMENTATION

Different activities require different segment lengths to be adequately detected. Experiments show that the higher the person's dynamics, the longer the segment needed to classify the activity precisely. Therefore, to improve the predictive accuracy of our hierarchical classifier with minimum segment length, we proposed adaptive segmentation. Different segment sizes for each parent node classifier are used, ensuring the size contains enough information to represent all child node classes (activities).

We start by selecting one second as a default segment length of 1 sec and then apply the Fast Fourier Transform (FFT) to the input segment. After that, the center frequency and bandwidth are examined by an adaptive segmentation function that increases the segment length to a maximum of two seconds according to the dynamics represented in that segment. The size decreases again to one second once the performed activity dynamics decline. So, each segment encloses enough information that can be used to distinguish different dynamics.

C. FEATURE EXTRACTION

For each of our local classifiers, different significant feature sets are selected by the decision tree to describe the sub-class variations better. The full features set includes time-domain, frequency-domain and IMU-based quasi-stationary inclination features. We proposed independent orientation features, including the magnitude of 3-D acceleration to distinguish between static and dynamic, which can be calculated using equation 5.

$$m = \frac{1}{N}\sum_{i=0}^{N-1}\sqrt{x_i^2 + y_i^2 + z_i^2} \tag{5}$$

where N is the segment length, x, y, and z are the sensor data's x-axis, y-axis and z-axis, respectively. The extracted time-domain features also include Root Mean Square (RMS) as given in equation 6, and the standard deviation of each sensor axis.

$$rms_j = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} j_i^2} \tag{6}$$

where $j \in \{x, y, z\}$

In addition, we used ten multiple percentile values to measure the distribution of the time-series signal. Percentiles are the values below which a certain percentage of the signal's information is found, which can be calculated using equation 7.

$$n = \frac{P}{100}N \tag{7}$$

where N is the number of values in the data set, P is the percentile, and n is the ordinal rank of a given value. Then, we transform the signal to the frequency domain using FFT to extract the frequency-domain feature vectors. The frequency-domain features mainly include FFT coefficients, spectral energy, spectral entropy, and power density. FFTs are excellent at analyzing vibration when there are a finite number of dominant frequency components, while Power Spectral Densities (PSD) are used to characterize random vibration signals.

PSD is the Fourier transform of the autocorrelation of the signal, which gives useful information when signals are contaminated by random noise, which is the case in IMU data. A PSD is computed by multiplying each frequency bin in an FFT by its complex conjugate and then normalizing it, making it valuable for random vibration analysis. The power spectral density function $X_{PSD}\,(f)$ is calculated from the Fourier transform $X(f)$ as in 8.

$$X_{PSD}\,(f) = lim_{\Delta f \to 0}\left[\frac{1}{2}\frac{X(f) * (f)}{\Delta f}\right] \tag{8}$$

where Δf is inverse of the segment duration calculated by equation 9.

$$\Delta f = \frac{1}{N} \tag{9}$$

In addition, we are using the IMU-based quasi-stationary inclination feature to differentiate between fine-grained activities by calculating the pitch angle for each sensor using equation 10. Fig. 8 shows the pitch angle for two ambiguous activities, walking and stairs. As explained earlier, "Climbing Stairs" activity usually includes an inclination fluctuation, which is clearly shown in the figure. As we will see later in the overall results, including the inclination (also known as pitch), orientation helps discriminate between these two ambiguous activities. IMU-based quasi-stationary inclination can be calculated by equation 10.

$$pitch = \frac{1}{N}\sum_{i=0}^{N-1} atan2\left(x_i, \sqrt{y_i^2 + z_i^2}\right) \tag{10}$$

where N is the segment length, x, y, and z are the x-axis, y-axis and z-axis of the sensor data, respectively.

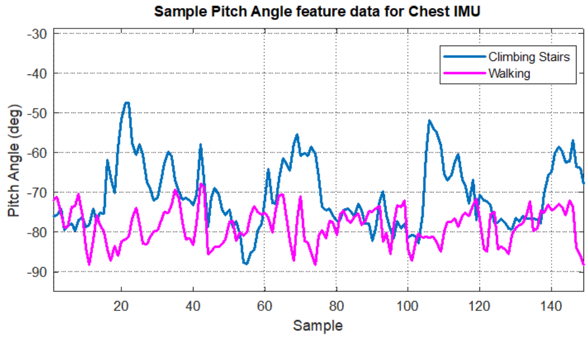


FIGURE 8. Typical pitch angle feature data for chest IMU for climbing stairs and walking activities.

D. POST PRUNING FOR CONSOLIDATING THE GLOBAL CLASSIFIER

Post-pruning of the decision tree (also called backward pruning) techniques are used to control the depth of the tree and overcome any possible overfitting by removing redundant insignificant local branches and feature-based rules of the tree after its construction. These local branches usually result in a minor data separation. Literature shows different fitness measures like Gini Impurity or Information Gain [57]. We used Percentage Coverage (PC) to measure the percentage of instances covered by a given rule, $PC\ (r)$ as in equation 11

$$PC\ (r) = \frac{\sum_{i=0}^{n} S_i(r)}{\sum_{i=0}^{n} S_i} \tag{11}$$

where n is the total number of subjects in the dataset, S_i is the set of instances of subject i and $S_i(r)$ is the set of instances of subject i covered by a rule r.

The main objective of using such a measure is to construct a more general decision tree that is not subject-dependent. So, a PC fitness measure is adopted with modifications to select the best small set of rules that boost the classification generalization and interpretation powers.

E. DATASET PARTITIONING

For using the data for the hierarchical framework, the data is first divided for the different local classifiers and then partitioned as training and testing sets. There are various ways to associate the subset training sets for each local classifier in the literature. We modified the "siblings" policy to define the set of positive (Tr^+) and negative (Tr^-) instances for the hierarchical training data for each class according to the following equations:

$$Tr^{+}\left(c_j\right) = T_p\left(c_j\right) \cup \sum_{i=0}^{k} T_p(c_{j.k}) \tag{12}$$

$$Tr^{-}\left(c_j\right) = \frac{Tr}{Tr^{+}\left(c_j\right)} \tag{13}$$

where $Tr^{+}\left(c_j\right)$ is the set of positive training examples of class c_j; $T_p\left(c_j\right)$is the set of true positive instances of class c_j; k number of child nodes of root class c_j; and $Tr^{-}\left(c_j\right)$ is the set of negative training examples of class c_j.

In evaluating the proposed AHDT system, we adopted the procedure of leave-one-subject-out data splitting. This method will overcome any accuracy estimation bias or classification overfitting problem to users/subjects, ensuring that the system can be used as a subject-independent classification system.

F. COMPLEXITY ANALYSIS

The computational complexity of the proposed AHDT system depends on the number of parent nodes in the tree taxonomy and the complexity of the base classifier, which is the decision tree classifier. The decision tree's complexity depends on two main factors. First is the number of features representing the input data's dimensionality—and second is the tree's depth, which determines the number of decision nodes and leaf nodes in the tree. Deeper trees can capture more complex relationships in the data, lowering the interpretability of the model and increasing computational complexity. Therefore, The decision tree's computational complexity during the training phase is $O(N\ M \log(N))$, where N is the number of dataset samples, and M is the number of features. The training phase consists of building multiple decision tree models equal to the number of parent nodes, K. Followed by post-pruning using the PC algorithm, whose complexity is defined as $O(M)$. Therefore, the proposed AHDT system time complexity is given by

$$\begin{aligned} AHDT_{complexity} &= O(K\ N\ M \log(N)) + O(M) \\ &= O(K\ N\ M \log(N)) \end{aligned}$$

while $K \ll M\ N$ So, it could be considered a constant scaling factor and can be ignored. Therefore, the proposed AHDT system time complexity during the training phase is

$$AHDT_{complexity} = O(N\ M \log(N)) \tag{14}$$

, which approximately equals the time complexity of training a single DT model. The space complexity of the AHDT system is $O(K\ N)$, which mainly depends on the tree depth,D, reflecting the model's global interpretability.

Hence, the complexity of the proposed AHDT system is much less than other enhanced tree-based classifiers like Random Forest(RF), Adaptive Boosting (AdaBoost)and Gradient Boosting Decision Trees (GBDT). For example, the random forest, which builds an ensemble of decision trees with a fixed number of trees, T, has space complexity $O(T\ N)$, where N is the number of samples [58]. Moreover, the computational complexity of RF training is generally proportional to the number of decision trees in the forest multiplied by the complexity of each tree. Therefore, the overall computational complexity of RF is approximate $O(T\ N\ M \log(N))$ [59]. It should be noted that the number of decision trees in the forest is generally much higher than the number of parent nodes in the case of AHDT.

In contrast, the computational complexity of training both AdaBoost and GBDT classifiers depends on several factors, including the number of weak learners (Decision Trees) in the ensemble, T, the number of samples, N, the number of features, M, and the complexity of each decision tree. Therefore, the overall computational complexity of GBDT and Adaboost is approximately $O(T\ N\ M \log(N))$.

The space complexity of those enhanced tree classifiers is generally higher than other classification algorithms like DT and RF. The space complexity is impacted by the number of iterations, T, the number of samples, N, and the number of features,M. Therefore, the space complexity of AdaBoost and GBDT is typically $O(T\ N\ M)$. These computational complexities will be analyzed experimentally in subsection IV-D. In addition, comparing the complexity of the AHDT system to deep learning shows the AHDT's superiority. Deep learning models are known for their computationally intensive nature due to their complex architecture and large number of parameters. Moreover, the computational complexity of deep learning models increases with the number of layers and nodes, making them both resource-intensive and time-consuming.

G. ACCURACY MEASURES

For evaluating the proposed AHDT system, we started with the commonly used flat classification measures for comparing our results to the literature. The percentage of relevant classification instances is called precision, $P(c_j)$. A second measure is the recall $R(c_j)$, which is the percentage of relevant total results correctly classified. We also calculated the harmonic mean of precision and recall, called the F1-score, $F1\left(c_j\right)$.

$$P(c_j) = \frac{TP\left(c_j\right)}{TP\left(c_j\right) + FP\left(c_j\right)} \tag{15}$$

$$R(c_j) = \frac{TP\left(c_j\right)}{TP\left(c_j\right) + FN\left(c_j\right)} \tag{16}$$

$$F1(c_j) = \frac{2P(c_j)R(c_j)}{P(c_j) + R(c_j)} \tag{17}$$

where $TP\left(c_j\right)$ is the number of True Positive instances of class c_j; $FP\left(c_j\right)$ is the number of False Positive of class c_j and $FN\left(c_j\right)$ is the number of False Negative of class c_j.

The accuracy measures have been modified to evaluate the hierarchical classification. The hierarchical accuracy represents the misclassification errors at the different levels of the classification hierarchy and considers hierarchical relationships between the classes [8]. Hierarchical measures aim to introduce different penalization for the misclassification rates at the different levels and paths of the hierarchy.

For example, in Figure 5, if the actual class is 2.3, since classes 2.2 and 2.3 are hierarchically closer to each other, sharing similar characteristics, wrongly classifying it as 2.2 should introduce less error than categorizing it as 1.1 or 1.2. It should also be considered that classifying at deeper (more specific) levels of the hierarchy is more challenging than at shallower (Less specific) ones.

Equations 18,19 and 20 show the calculation of hierarchical precision (hP), hierarchical recall (hR) and hierarchical F1-score ($hF1$) respectively.

$$hP(c_j) = \frac{TP\left(c_j\right) + \sum_{i\in pr(j)} TP(c_i)}{TP\left(c_j\right) + FP\left(c_j\right) + \sum_{i\in pr(j)} \left(TP\left(c_i\right) + FP(c_i)\right)} \tag{18}$$

$$hR(c_j) = \frac{TP\left(c_j\right) + \sum_{i\in pr(j)} TP(c_i)}{TP\left(c_j\right) + FN\left(c_j\right) + \sum_{i\in pr(j)} \left(TP\left(c_i\right) + FN(c_i)\right)} \tag{19}$$

$$hF1(c_j) = \frac{2hP(c_j)hR(c_j)}{hP(c_j) + hR(c_j)} \tag{20}$$

where $TP\left(c_j\right)$ is the number of true positive instances of class c_j; $FP\left(c_j\right)$ is the number of false positive instances of class c_j and $FN\left(c_j\right)$ is the number of false negative instances of class c_j.

Another way to assess the classifier model's prediction performance is by plotting the receiver operating characteristic (ROC) curve. ROC curve is a visualization tool for comparing classifiers, and it graphs the true positive rate as a function of the false positive rate for all possible decision probability thresholds. The lower the threshold, the higher the true positive rate and the higher the false positive rate. Therefore, the closer the ROC curve is to the top-left corner, the higher the discriminatory ability of the classifier. The nearer the curve to the diagonal line, the closer the classifier performance is to a random guessing classifier. In the case of visualizing ROC for the multiclass classifier, we are using the one-versus-all scheme, which compares each class as a positive class against all remaining classes as one consolidated negative class.

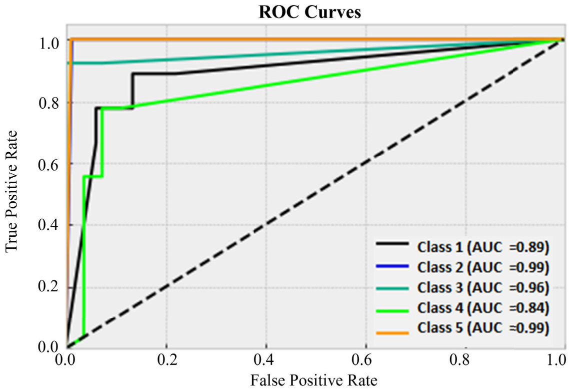


FIGURE 9. Example of Receiver Operating Characteristic (ROC) graph for multiclass classifier using the one vs all method.

Further, the Area under the ROC curve (AUC) is a measure that outlines the ROC in single numbers proportional to the discriminatory ability of classifiers to rank different samples correctly. It is a statistic representing the probability of correct prediction of a randomly selected sample of the positive class against the other class (classes). The higher the AUC, the better the ability of the model to distinguish the given class from others. For example, in Fig. 9, the AUC has the interpretation that the model classifies randomly chosen samples from class 1 correctly with a probability of 89% while classifying randomly chosen samples from class 5 with a probability of 99%.

IV. EXPERIMENTS

In this section, we benchmark the proposed adaptive hierarchical HAR model, denoted as AHDT, to compare with other state-of-the-art methods on six publicly available datasets. In addition, we investigate the method's interpretability and computational complexity.

A. EXPERIMENTAL SETUP

For the evaluation of the proposed model, we compared the performance and interpretability of our hierarchical method to four well-known decision tree-based flat classifiers: flat Decision Tree (DT), Random Forest (RF), Adaptive Boosting (AdaBoost) and Gradient Boosting Decision Tree (GBDT). We used the same base estimator with the same hyperparameter tuning for all classifiers. Each base DT classifier is trained with specific training parameters, including a measure of entropy, the maximum depth of the subtree, and the minimum number of samples necessary to split an internal node. Entropy is utilized as a measure of impurity for the split, while a maximum tree depth of 5 is implemented for constructing less complex trees. Furthermore, the minimum number of samples required to split an internal node is 5% of the total training samples.

TABLE 2. Human activities contained in different datasets.

Activities	REALDISP	DaLiAc	MHEALTH	PAMAP2	WISDM	HARTH
Standing		✓	✓	✓	✓	✓
Sitting		✓	✓	✓	✓	✓
Lying		✓	✓	✓		✓
Jogging	✓		✓		✓	
Free Walking	✓	✓	✓	✓	✓	✓
Nordic Walking				✓		
Free Running	✓		✓	✓		
Treadmill running		✓				
Ergometer (50W)		✓				
Egometer (100W)		✓				
Climbing Stairs			✓			
Ascending stairs		✓		✓	✓	✓
Descending stairs		✓		✓	✓	✓
WaistBendsForward			✓			
Arms frontal elevation	✓		✓			
Arms lateral elevation	✓					
Frontal hand claps	✓					
Arms frontal crossing	✓					
Shoulders high amplitude rotation	✓					
Shoulders low amplitude rotation	✓					
Knees to Beast	✓					
Heels to Backside	✓					
Knees bending Forward	✓					
Rotation on Knees	✓					
Rowing	✓					
Arms inner rotation	✓					
crouching	✓		✓			
Free Bicycling	✓		✓			
Elliptical bike	✓					
Cycling				✓		
Cycling Sit						✓
Cycling Stand						✓
Jumping Front & Back	✓		✓			
Jumping Sideways	✓					
Jumping jack	✓					
Jumping Up & Down	✓					
Jumping rope	✓	✓		✓		
Washing dishes		✓				
Ironing				✓		
Vacuuming		✓		✓		
Sweeping		✓				
Arms Outstretch	✓					
Elbows Bent	✓					
Waist bends forward	✓					
Waist rotation	✓					
Reach heels backwards	✓					
Lateral bend to Left & Right	✓					
Lateral bend arms up	✓					
Forward stretching	✓					
Opposite twist	✓					
Waist bends opposite direction	✓					

During the training of these classifiers, we used our presented full feature set with fixed segmentation of window size of 2.5 seconds for all flat classification methods and adaptive segmentation for our proposed hierarchical method as described in III-B. In this research, we considered only the accelerometer sensor data. The proposed HAR system was developed using Python 3.7 with Scikit-learn and NumPy libraries. The system was trained and tested on an Asus laptop i7-intel CPU running at 2.8 GHz with an Nvidia GTX 1050 GPU with 8 GB RAM.

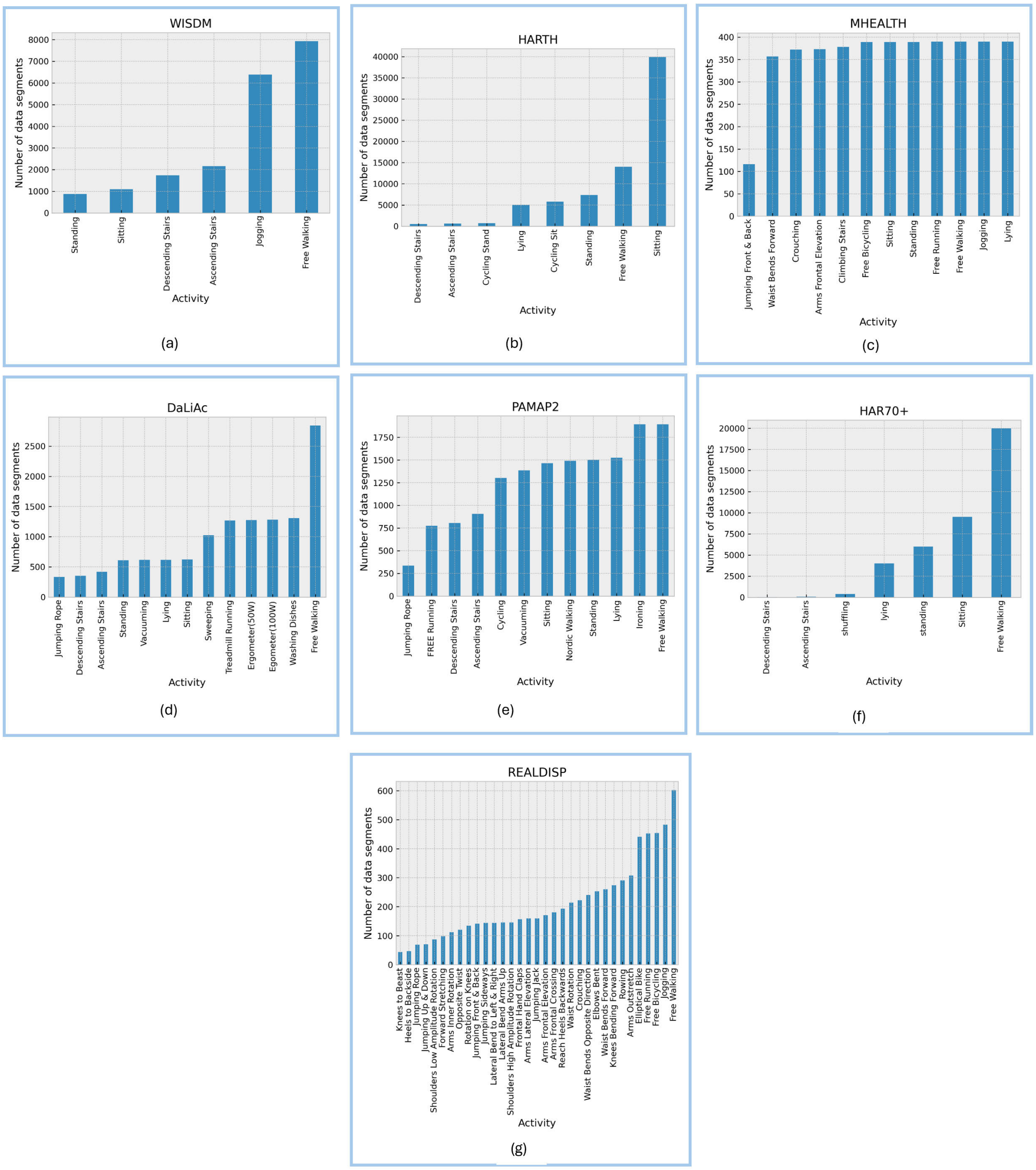


FIGURE 10. **Graphical representation of the activities data distributions in the different datasets:(a) WISDM Dataset, (b) HARTH Dataset, (c) MHEALTH Dataset, (d) DaLiAc Dataset, (e) PAMAP2 Dataset, (f) HAR70+ Dataset, (g) REALDISP Dataset.**

B. DATASETS

The system was tested on six publicly available datasets, with a total of about fifty different human activities summarized in Table 2. The distribution of the activity data in the different datasets is shown in Fig. 10. Some activities were renamed for a better understanding of their description. The datasets are:

1) The Wireless Sensor Data Mining (WISDM) Dataset [60]: The WISDM dataset consists of tri-axis accelerometer data with a sampling frequency of 20Hz collected from 29 subjects performing six activities, which are walking, climbing stairs ascending, climbing stairs descending, sitting, standing, and lying with the imbalanced distribution. This data was collected using several types of Android phones placed in the front pants-leg pocket. The advantage of using this dataset is that the collected data is platform-independent. It has the smallest number of activities among the different datasets. For this dataset with a small number of activities, a simple tree taxonomy with two levels can be used, as shown in Fig. 11.
2) The Human Activity Recognition Trondheim (HARTH) dataset [61]: The HARTH dataset used two tri-axial Axivity AX3 accelerometers for data acquisition with a sampling rate of 50 Hz. Twenty-two healthy adults' data were collected while performing the

activities: sitting, standing upright, lying, walking in all directions, running, climbing stairs ascending, climbing stairs descending, cycling while sitting, and cycling while standing while riding a bike. The sensors were attached to each subject's right front thigh and lower back. The axes alignment of the sensors is known. The lower back sensor's x-axis points downward, the y-axis to the left, and the z-axis forward. The thigh sensor's y-axis points to the right and the z-axis backward. The activities were labelled by experts utilizing the camera's video signal. The distribution of the activity data is significantly imbalanced.

3) The Mobile Health (MHEALTH) dataset [28]: The MHEALTH dataset comprises body motion and vital signs recordings for ten subjects performing 12 physical activities collected using three Shimmer2 IMU wearable sensors placed on the subject's chest, right wrist and left ankle. This dataset was gathered in an out-of-lab environment with no constraints with a sampling rate of 50 Hz.

4) The Physical Activity Monitoring for Aging People (PAMAP2) dataset [62]: The PAMAP2 dataset contains mainly data from twelve physical activities performed by nine subjects wearing three inertial measurement units with a sampling frequency of 100Hz. The IMU units were positioned over the wrist on the dominant arm, on the chest and on the dominant side's ankle. The activities include lying, sitting, standing, free walking, running, cycling, nordic walking, ascending stairs, descending stairs, vacuuming, ironing, folding laundry, house cleaning, playing soccer, and jumping rope.

5) The Daily Life Activities (DaLiAc) dataset [63]: The DaLiAc dataset consists of data from 19 subjects performing 13 daily life activities. The activities are sitting, laying, standing, washing dishes, vacuuming, sweeping, walking, climbing stairs up, climbing stairs down, treadmill running, cycling on an ergometer (50W), cycling on an ergometer (100W), and jumping. The data was collected using four Shimmer2 wearable sensors placed on the right hip, chest, right wrist, and left ankle with known coordinate axes orientation of each sensor node. The sampling rate was set to 204.8 Hz. This dataset is heavily imbalanced, containing about 3000 samples for one activity and less than 500 samples for another.

6) The REAListic sensor DISPlacement (REALDISP) dataset [64]: The REALDISP includes a wide range of physical activities of whole-body motion and body part-specific activities focused on the trunk, upper extremities, and lower extremities. The activities are walking, jogging, free running, jumping up and down, jumping front and back, jumping sideways, jumping jack, jumping rope, arms outstretched, elbows bent, waist bends forward, waist rotation, waist bends opposite direction, reaching heels backwards, lateral bend to left and right, lateral bend arms up, forward stretching, opposite twist, arms lateral elevation, arms frontal elevation, frontal hand claps, arms frontal crossing, shoulders high amplitude rotation, shoulders low amplitude rotation, arms inner rotation, knees to the beast, heels to backside, crouching, knees bending forward, rotation on knees, rowing, elliptical bike, free bicycling. This dataset was collected using nine Xsens IMUs distributed on the subject's body and with a sampling rate of 50 Hz. In this research, we used five sensor data attached to the subject's right lower arm, right upper arm, back, right calf, and right thigh.

7) The HAR70 dataset [65] comprises data collected from 18 elderly individuals ranging from 70 to 95 years old. Among these participants, four consistently relied on a walker for ambulation, while one used a walking stick for outdoor activities. Throughout the study, each person wore two accelerometers and a chest-mounted camera and completed multiple repetitions of typical daily activities such as walking, standing, sitting, and lying down. The Axivity AX3 accelerometers were placed on the lower back and right thigh and sampled at 50 Hz, with a range of ±8 g.

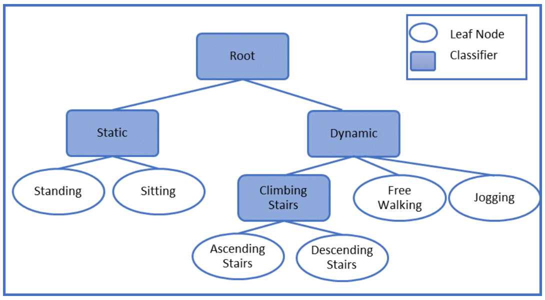


FIGURE 11. Tree taxonomy of WISDM dataset for hierarchical classification using the AHDT system.

C. EXPERIMENTAL RESULTS AND DISCUSSION

In this subsection, we present the results and discuss the performance of the proposed AHDT system against conventional classifiers. In the first experiment, comparing the F1-score on the WISDM dataset as shown in Table 3 shows that our proposed system AHDT outperforms all the other tree-based algorithms for all activities. A deeper look at the different classifiers' performance in distinguishing the activities, Ascending Stairs and Descending Stairs, shows all other classifiers F1-score is about or less than 0.5, which is worse than a random guessing classifier. The similarity between these two activities and the data collection using different platforms makes such a task harder for the other classifiers. This can be verified by looking into the RF and GBDT confusion matrix as in Fig. 12 and Fig. 13 respectively. The results show both RF and GBDT classifiers are highly misclassifying the climbing stairs activities with each other and with free walking activity.

TABLE 3. Comparison of classifiers' performance using WISDM dataset.

‘	DT			RF			AdaBoost			GBDT			AHDT		
Activity	P	R	F1	P	R	F1	P	R	F1	P	R	F1	hP	hR	hF1
Sitting	0.95	0.9	0.93	0.96	0.92	0.94	0.99	0.95	0.97	0.96	0.92	0.94	0.99	1	0.99
Standing	0.89	0.95	0.92	0.91	0.95	0.93	0.94	0.99	0.97	0.93	0.95	0.94	1	0.99	0.99
Free Walking	0.74	0.88	0.81	0.86	0.8	0.83	0.84	0.69	0.76	0.87	0.83	0.85	0.95	0.95	0.95
Ascending Stairs	0.38	0.25	0.3	0.45	0.54	0.49	0.36	0.49	0.42	0.48	0.54	0.51	0.96	0.91	0.93
Descending Stairs	0.33	0.18	0.23	0.56	0.53	0.54	0.38	0.51	0.43	0.55	0.55	0.55	0.94	0.94	0.94
Jogging	0.93	0.92	0.93	0.93	0.94	0.93	0.9	0.91	0.9	0.92	0.94	0.93	0.94	0.96	0.95

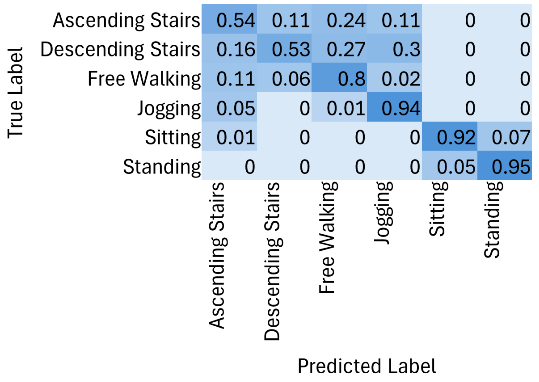


FIGURE 12. Confusion matrix for the Random Forest classifier using WISDM dataset.

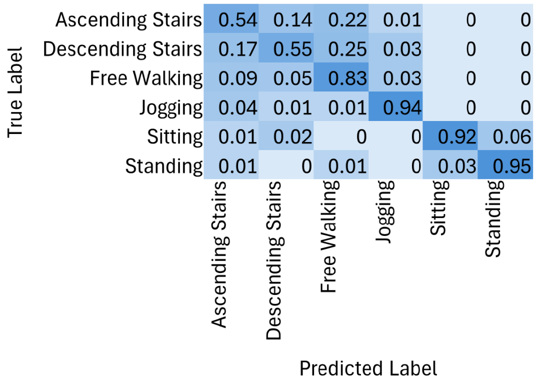


FIGURE 13. Confusion matrix for the Gradient Boosting Decision Tree based Classifier using WISDM dataset.

On the other hand, examining the hierarchical confusion matrices of our proposed hierarchical system for the same dataset shown in Fig. 14 shows that the AHDT system has higher separability abilities. This demonstrates the ability of AHDT to detect the significant features that distinguish between complex and hard-to-classify activities.

Another experiment was performed on the HARTH dataset. The HARTH dataset contains two more ambiguous activities than WISDM, resulting in one more LCPN to classify these two similar activities. However, it uses only one platform(sensor set) for data collection, with extra sensor data collected from additional sensor placement. This configuration could make the data less ambiguous than WISDM. Table 4 shows the F1 score for the dataset. Our AHDT system outperforms the other classifiers on five activities and performs similarly on the remaining activities except the Cycle Sit activity when RF had the best results.

The RF classifier's better performance in this data set could be due to some factors described as follows: First, the size of this dataset is ten times the previous one, which results in a much larger dataset. Second, the uniform platform data collection introduced fewer variant sensor errors in the data, which could result in a much easier classification problem. However, higher dynamic activities, like climbing stairs, Ascending and descending, and Cycling while Sitting and Standing, is still point of weakness for all other enhanced tree-based classifiers.

Analyzing the confusion matrix shown in Fig. 15 and the ROC curve of it, which is shown in Fig. 16, shows that 15% of the Cycling activities data is misclassified as Free walking with a still high probability of 96%. This reflects the scalability of the AHDT system, which can be further proved by studying the Confusion Matrices and ROC curves of the Climbing Stairs and Cycling activities, shown in figures 17, 18, 19, 20, respectively. The AUC ROC of these four confusing activities shows a high discriminative ability range from 91%-99%. As a result, exploring the ADHT with the HARTH dataset showed the stability and robustness of the system with more elevated classification performance.

In comparing the MHEALTH dataset to the previous two datasets, it is a balanced dataset with additional sensor data and a reasonable number of samples, making it less challenging. Analyzing the F1 score for the different classifiers, shown in Table 5, reveals comparable performance between our AHDT to both RF and Adaboost systems. This performance similarity raises a question about which classification model we should use for less challenging problems. One could look at other factors like the computational cost of each one of the systems, which will be discussed in section IV-D.

In the case of the PAMAP2 dataset, although it has a much higher number of training samples than MHEALTH, the class imbalance adds an extra barrier to the classification problem. The comparison between F1 scores in Table 6 reveals that AHDT surpassed the other methods in almost all activities. These results demonstrate our AHDT system's ability to manage balanced and imbalanced data classification effectively by employing the shared characteristics and inherited relations among the classes.

TABLE 4. Comparison of classifiers' performance using HARTH dataset.

‘	DT			RF			AdaBoost			GBDT			AHDT		
Activity	P	R	F1	P	R	F1	P	R	F1	P	R	F1	hP	hR	hF1
Sitting	0.99	0.99	0.99	0.99	1	1	0.99	0.99	0.99	0.99	1	0.99	1	0.98	0.99
Lying	1	0.98	0.99	1	1	1	1	0.98	0.99	1	0.98	0.99	1	1	1
Standing	0.95	0.94	0.94	0.98	0.94	0.96	0.93	0.93	0.93	0.97	0.95	0.96	1	1	1
Free Walking	0.88	0.96	0.92	0.94	0.98	0.96	0.87	0.88	0.87	0.93	0.94	0.94	0.99	0.92	0.95
Ascending Stairs	0	0	0	0.79	0.4	0.53	0.68	0.31	0.43	0.28	0.46	0.35	0.97	0.96	0.96
Descending Stairs	0	0	0	0.85	0.51	0.64	0.14	0.49	0.22	0.41	0.58	0.48	0.96	0.97	0.96
Cycling Sit	0.95	0.76	0.84	0.96	0.97	0.97	0.95	0.85	0.9	0.96	0.9	0.93	0.96	0.87	0.91
Cycling Stand	0.23	0.31	0.26	0.75	0.73	0.74	0.8	0.62	0.7	0.41	0.2	0.27	0.9	0.89	0.89

TABLE 5. Comparison of classifiers' performance using MHEALTH dataset.

‘	DT			RF			AdaBoost			GBDT			AHDT		
Activity	P	R	F1	P	R	F1	P	R	F1	P	R	F1	hP	hR	hF1
Sitting	1	0.8	0.89	1	0.9	0.95	0.96	0.8	0.87	0.85	0.7	0.77	1	1	1
Lying	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Standing	0.83	1	0.91	0.91	1	0.95	0.83	0.97	0.89	0.9	0.9	0.9	1	1	1
Free Walking	1	1	1	1	1	1	1	1	1	0.97	0.97	0.97	0.98	0.98	0.98
Climbing Stairs	0.99	0.99	0.99	1	1	1	1	1	1	0.83	0.96	0.89	0.99	0.98	0.99
Waist Bends Forward	0.94	1	0.97	1	1	1	1	1	1	0.99	0.98	0.99	1	1	1
Arms Frontal Elevation	0.99	0.99	0.99	1	1	1	1	1	1	1	0.99	0.99	1	1	1
Crouching	0.99	0.95	0.97	1	1	1	1	1	1	0.86	0.98	0.91	0.99	1	0.99
Free Bicycling	0.99	0.98	0.99	1	1	1	1	1	1	1	0.92	0.96	1	0.99	0.99
Jogging	0.85	0.91	0.88	0.83	0.95	0.89	0.86	0.92	0.89	0.83	0.94	0.88	0.98	0.98	0.98
Free Running	0.91	0.84	0.87	0.94	0.82	0.87	0.91	0.85	0.88	0.94	0.82	0.88	0.98	0.98	0.98
Jumping Front & Back	1	0.99	1	1	0.98	0.99	1	1	1	0.95	0.9	0.92	1	0.96	0.98

TABLE 6. Comparison of classifiers' performance using PAMAP2 dataset.

‘	DT			RF			AdaBoost			GBDT			AHDT		
Activity	P	R	F1	P	R	F1	P	R	F1	P	R	F1	hP	hR	hF1
Sitting	0.63	0.53	0.57	0.83	0.78	0.8	0.66	0.67	0.66	0.71	0.63	0.67	0.86	0.88	0.87
Lying	0.97	0.79	0.87	0.93	0.94	0.94	0.99	0.73	0.84	0.89	0.91	0.9	0.96	0.91	0.93
Standing	0.55	0.57	0.56	0.74	0.73	0.73	0.71	0.73	0.72	0.7	0.66	0.68	0.87	0.86	0.86
Vacuuming	0.47	0.65	0.54	0.65	0.88	0.75	0.64	0.84	0.73	0.78	0.86	0.82	0.91	0.95	0.93
Free Walking	0.53	0.61	0.57	0.72	0.79	0.75	0.62	0.68	0.64	0.72	0.88	0.8	0.87	0.84	0.85
Ascending Stairs	0.57	0.53	0.55	0.7	0.84	0.76	0.61	0.72	0.66	0.71	0.81	0.76	0.86	0.87	0.87
Descending Stairs	0.59	0.47	0.52	0.9	0.67	0.77	0.71	0.7	0.71	0.71	0.73	0.72	0.86	0.84	0.85
Jumping Rope	0.54	0.35	0.43	0.84	0.39	0.53	0.69	0.3	0.42	0.81	0.3	0.44	0.78	0.97	0.86
Cycling	0.72	0.65	0.68	0.92	0.83	0.87	0.88	0.76	0.82	0.95	0.84	0.89	0.92	0.83	0.87
FREE Running	0.92	0.39	0.55	0.83	0.71	0.76	0.88	0.54	0.67	0.9	0.62	0.73	0.96	0.83	0.89
Nordic Walking	0.58	0.48	0.52	0.92	0.65	0.76	0.79	0.55	0.65	0.87	0.69	0.77	0.89	0.82	0.85
Ironing	0.61	0.86	0.71	0.77	0.9	0.83	0.66	0.89	0.76	0.73	0.89	0.81	0.92	0.89	0.9

In addition, the results of the experiment on the DaLiAc dataset are shown in Table 7. The DaLiAc dataset, which has almost double the number of samples than the PAMAP2 dataset with an extra source of information, the subject's hip sensor data, which theoretically gives the classification algorithm a higher opportunity to define separability between the classes and, therefore, more precise classification performance. Nevertheless, all other classifiers have less accurate F1-score measures than our AHDT for most activities. The most impacted performance is distinguishing the ambiguous activities Egometer(50w) and Egometer(100w). By examining the confusion matrix of the AdaBoost algorithm on this dataset, for example, shown in Fig. 21, and comparing that to the confusion matrix of the Bicycling LCPN component of the AHDT, shown in Fig. 22, it is clear that the percentage of truly classified Egometer(50w) activity samples increased from 48%, in case of AdaBoost, to 86% in case of the AHDT system. Undoubtedly, the profound comprehensive activity recognition of the AHDT overcomes this issue by giving much higher performance while sustaining the lower classification complexity.

Examining the REALDISP dataset, which has the most elevated number of activities, requires using the entire taxonomy hierarchy shown in Fig. 6. As an outgrowth, theoretically, increasing the number of activities will extend the number of LCPN ensemble classifiers of the proposed AHDT system and, therefore, raise the complexity of our AHDT

TABLE 7. Comparison of classifiers' performance using DaLiAc dataset.

‘	DT			RF			AdaBoost			GBDT			AHDT		
Activity	P	R	F1	P	R	F1	P	R	F1	P	R	F1	hP	hR	hF1
Sitting	0.94	0.85	0.89	0.94	0.95	0.95	0.96	0.92	0.94	0.93	0.86	0.89	0.99	1	0.99
Lying	1	1	1	1	1	1	1	1	1	1	0.95	0.97	1	1	1
Standing	0.83	0.81	0.82	0.87	0.89	0.88	0.88	0.9	0.89	0.84	0.9	0.87	0.99	1	0.99
Washing Dishes	0.91	0.94	0.92	0.94	0.93	0.93	0.98	0.95	0.96	0.95	0.96	0.95	0.93	0.91	0.92
Vacuuming	0.8	0.64	0.71	0.86	0.75	0.8	0.8	0.71	0.75	0.83	0.77	0.8	0.92	0.92	0.92
Sweeping	0.76	0.87	0.81	0.85	0.93	0.89	0.79	0.9	0.84	0.85	0.93	0.88	0.97	0.95	0.96
Free Walking	0.96	0.97	0.97	0.98	0.98	0.98	0.97	0.97	0.97	0.98	0.98	0.98	0.99	0.99	0.99
Ascending Stairs	0.77	0.86	0.81	0.87	0.91	0.89	0.86	0.88	0.87	0.92	0.89	0.9	0.95	0.93	0.94
Descending Stairs	0.85	0.8	0.83	0.91	0.82	0.87	0.83	0.81	0.82	0.94	0.91	0.92	0.95	0.93	0.94
Treadmill Running	1	0.95	0.97	1	1	1	1	1	1	1	0.99	0.99	0.98	0.98	0.98
Ergometer(50W)	0.56	0.45	0.5	0.64	0.6	0.62	0.61	0.48	0.54	0.65	0.64	0.65	0.93	0.91	0.92
Egometer(100W)	0.54	0.66	0.59	0.62	0.67	0.64	0.58	0.7	0.63	0.65	0.66	0.66	0.9	0.94	0.92
Jumping Rope	0.99	1	1	1	1	1	1	1	1	0.95	0.99	0.97	0.97	0.99	0.98

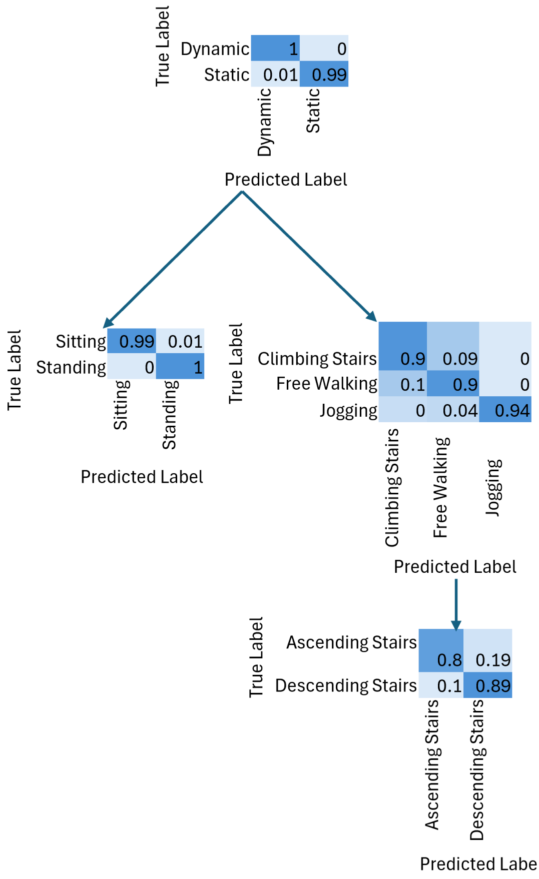


FIGURE 14. Hierarchical confusion matrix of the proposed classifier using WISDM dataset.

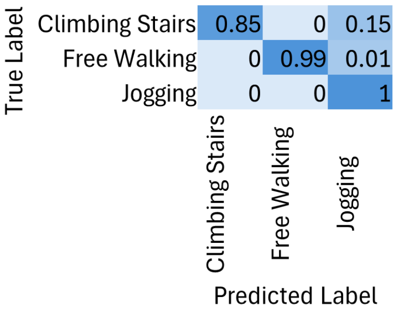


FIGURE 15. Confusion matrix of the Dynamic LCPN component of the proposed model using HARTH dataset.

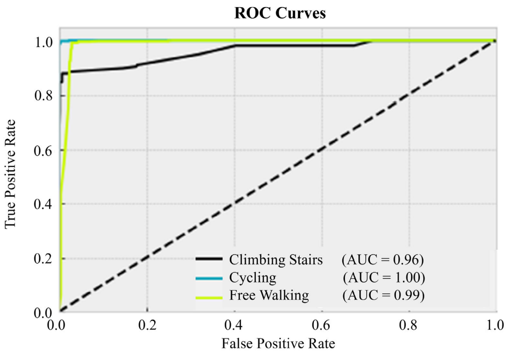
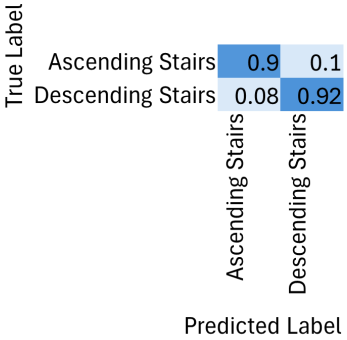


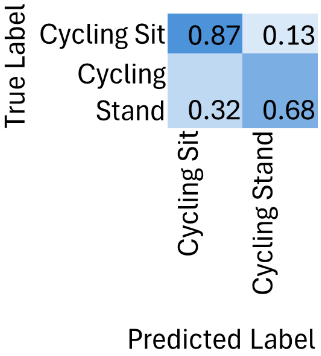
FIGURE 16. ROC curve of the Dynamic LCPN component of the proposed model using the HARTH dataset.

system. In this case, the AHDT trained about nineteen LCPN ensembles, while RF trained about one hundred ensembles, and both Adaboost and GBDT trained much more than that. Noting that all of these classifiers share the same DT ensemble's hyperparameters proves that AHDT diminishes the system's computational complexity than the other enhanced tree-based classifiers with the growing number of classes. At the same time, studying the F1-measure results, shown in table8, demonstrates that our proposed AHDT system helps distinguish between the utmost number of activities that can cause ambiguity in classification. The AHDT classification performance surpasses the other classifiers' performance without sacrificing the interpretability while maintaining the practicality of design and lower complexity. For example, the LCPN component that distinguishes between the free running and jogging activities uses a straightforward tree of one node depending on the distribution of the pitch angle of the z-axis of the accelerometer sensor attached to the right thigh of the subject who is performing the activity as shown in Fig. 23.



	Ascending Stairs	Descending Stairs
Ascending Stairs	0.9	0.1
Descending Stairs	0.08	0.92

FIGURE 17. Confusion matrix of Climbing Stairs LCPN component of the proposed AHDT system using HARTH dataset.



	Cycling Sit	Cycling Stand
Cycling Sit	0.87	0.13
Cycling Stand	0.32	0.68

FIGURE 18. Confusion matrix of Cycling LCPN component of the proposed model using HARTH dataset.

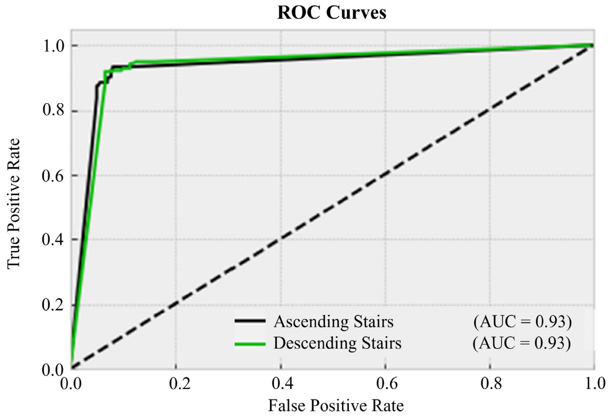


FIGURE 19. ROC curve of Climbing Stairs LCPN component of the proposed model using HARTH dataset.

Another example of the high interpretability of the proposed AHDT system is the LCPN component, which distinguishes between four higher complex physical exercises, which are waist bends forward, waist rotation, reaching heels backwards, and waist bends while reaching the foot with the opposite hand. This LCPN component depends on only three features: the frequency bandwidth of the z-axis of the right arm's accelerometer sensor, Root Mean Square(RMS) and the second maximum of the x-axis of the accelerometer sensor attached to the back of the subject as shown in Fig. 24. This elevated interpretability helps detect the origins of any misclassification and gives much more increased confidence in the system output.

To study whether the proposed AHDT system could be adopted for monitoring activities for specific populations, such as older people or those with physical disabilities, we tested The HAR70 dataset. The HAR70 dataset primarily

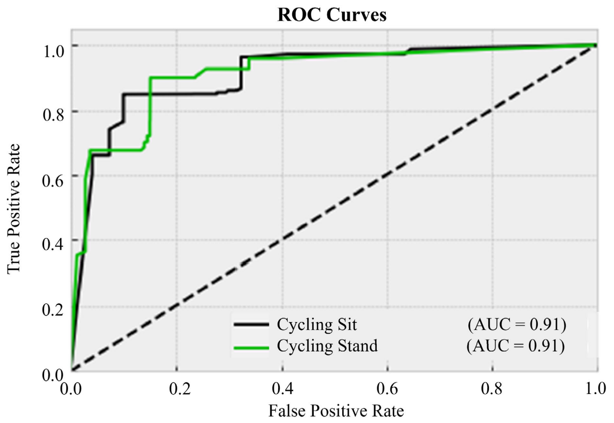
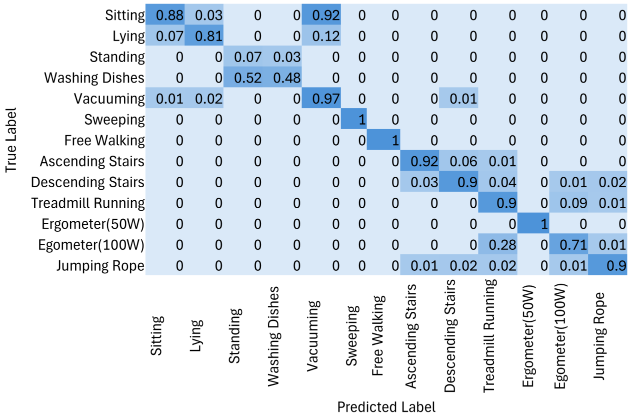
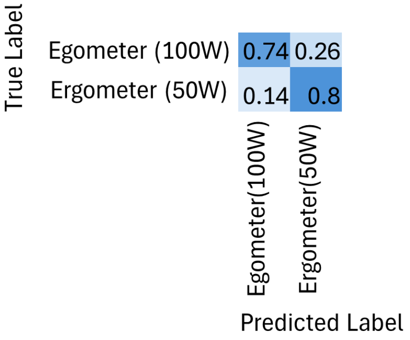


FIGURE 20. ROC curve of Cycling LCPN component of the proposed model using HARTH dataset.



	Sitting	Lying	Standing	Washing Dishes	Vacuuming	Sweeping	Free Walking	Ascending Stairs	Descending Stairs	Treadmill Running	Ergometer(50W)	Egometer(100W)	Jumping Rope
Sitting	0.88	0.03	0	0	0.92	0	0	0	0	0	0	0	0
Lying	0.07	0.81	0	0	0.12	0	0	0	0	0	0	0	0
Standing	0	0	0.07	0.03	0	0	0	0	0	0	0	0	0
Washing Dishes	0	0	0.52	0.48	0	0	0	0	0	0	0	0	0
Vacuuming	0.01	0.02	0	0	0.97	0	0	0	0.01	0	0	0	0
Sweeping	0	0	0	0	0	1	0	0	0	0	0	0	0
Free Walking	0	0	0	0	0	0	1	0	0	0	0	0	0
Ascending Stairs	0	0	0	0	0	0	0	0.92	0.06	0.01	0	0	0
Descending Stairs	0	0	0	0	0	0	0	0.03	0.9	0.04	0	0.01	0.02
Treadmill Running	0	0	0	0	0	0	0	0	0	0.9	0	0.09	0.01
Ergometer(50W)	0	0	0	0	0	0	0	0	0	0	1	0	0
Egometer(100W)	0	0	0	0	0	0	0	0	0	0.28	0	0.71	0.01
Jumping Rope	0	0	0	0	0	0	0	0.01	0.02	0.02	0	0.01	0.9

FIGURE 21. Confusion matrix of the AdaBoost classifier using DaLiAc dataset.



	Egometer(100W)	Ergometer(50W)
Egometer (100W)	0.74	0.26
Ergometer (50W)	0.14	0.8

FIGURE 22. Confusion matrix of Bicycling LCPN component of the proposed model using DaLiAc dataset.

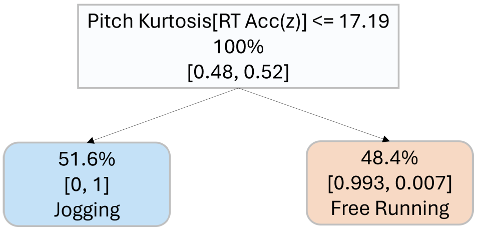


FIGURE 23. Tree derived from Running LCPN component of the proposed model using REALDISP dataset.

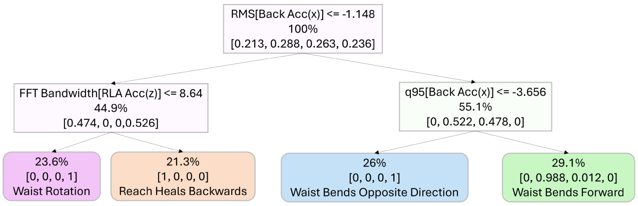


FIGURE 24. Tree derived from Waist Movement LCPN component of the proposed model using REALDISP dataset.

consists of walking activity samples, which make up more than 50% of the total dataset. The remaining samples are for static activities such as sitting, standing, and lying down, which account for nearly half of the dataset. However, physically demanding activities like climbing stairs and shuffling only make up less than 2% of the dataset, comprised mainly of walking activity samples, accounting for over 50% of the total dataset. The remaining samples are for static activities such as sitting, standing, and lying down, taking up nearly half of the dataset. However, the dataset only contains less than 2% of the samples for physically demanding activities such as climbing stairs and shuffling.

This uneven distribution of data makes the classification problem difficult to model and increases the risk of overfitting the dominant class. As shown in Table 9. Shuffling, ascending stairs, and descending stairs activity prediction rate by all other tree-based classifiers is worse than that of random classifiers. In comparison, AHDT achieved F1-score of 0.84, 0.79 and 0.79 for these activities, respectively.

In addition, the prediction performance of the proposed AHDT of shuffling and walking activities is much higher and can't be compared to other systems. In this dataset, these activities have subtle differences in movements due to the nature of the limited mobility of the subject older people. This demonstrates the effectiveness of our proposed system in accurately predicting activities with subtle movement differences.

D. COMPUTATIONAL COMPLEXITY

In this subsection, we are comparing the computational complexities experimentally by measuring the CPU processing time to execute the training phase of the different algorithms. Table 10 lists the actual CPU processing time in minutes taken by each algorithm in generating the classification model of each dataset. Evidently, the processing time of the AHDT algorithm is much less than other tree-based enhanced algorithms, namely, RF, AdaBoost and GBDT classifiers. In the worst scenario, the processing time of the proposed system is about half the time the fastest algorithm takes. For example, the processing time taken by the AHDT System to generate the classification model of the DaLiAc dataset is about 73% of the time taken by the RF algorithm for the same dataset. In contrast, the AHDT takes about 1% of the time needed to build a GBDT-based classification model on the same dataset.

In addition, the CPU processing time of building a model for the WISDM, which is a larger dataset with fewer activities compared to the DaLiAc dataset, increased rapidly for the other methods. For example, it is increased by about 700% for the RF classifier and about 150% for GDBT. At the same time, it is less by 26% for the proposed AHDT.To elaborate, regardless of the input dataset size, the AHDT system generates an accurate result efficiently.

E. ADVANTAGES AND CHALLENGES OF THE AHDT SYSTEM

The AHDT system has several advantages. One of its key advantages is its ability to provide an easily interpretable visual representation that does not require a deep understanding of complex black box models. Observing the decision path outlined in the tree allows one to readily determine the reasoning behind specific classification outputs, thereby simplifying the identification process of classification errors. This interpretability feature is valuable as it ensures that the results are accurate and reliable.

Moreover, the AHDT system is designed to select important features and disregard less significant ones, preventing overfitting. This process helps reveal each feature's impact and facilitates feature engineering. Also, extending the AHDT system to handle multi-label classification tasks with confidence levels is possible and could be an interesting avenue for future research. This extension could further enhance the system's capabilities and expand its potential applications.

On the other hand, the AHDT could have some challenges. The AHDT system relies on a DT algorithm that employs a greedy methodology for data splitting at each node. It selects the feature that yields the most significant reduction in impurity or error at that stage. While this approach is effective in many cases, it is not guaranteed to produce the most optimal tree structure.

Furthermore, the AHDT requires a predetermined tree taxonomy to accurately reflect the inherent hierarchical relationships between classes. While this requirement may be considered a system limitation, it presents an exciting avenue for future investigation.

TABLE 8. Comparison of classifiers' performance using REALDISP dataset.

‘	DT			RF			AdaBoost			GBDT			AHDT		
Activity	P	R	F1	P	R	F1	P	R	F1	P	R	F1	hP	hR	hF1
Free Walking	0.93	0.9	0.91	0.97	1	0.99	1	1	1	0.62	0.93	0.74	0.99	0.96	0.97
Jumping Rope	0.68	0.65	0.67	0.77	0.93	0.84	0.75	0.81	0.78	0.64	0.71	0.68	0.97	0.99	0.98
Waist Bends Forward	0.89	0.84	0.87	0.99	0.99	0.99	0.99	1	0.99	0.98	0.91	0.94	1	0.97	0.98
Arms Frontal Elevation	0.46	0.66	0.55	0.88	0.89	0.89	0.81	0.88	0.84	0.97	0.85	0.91	1	0.92	0.96
Crouching	0.8	0.41	0.54	0.91	0.91	0.91	0.9	0.99	0.94	0.86	0.77	0.81	1	1	1
Free Bicycling	0.98	0.91	0.94	1	1	1	1	0.97	0.98	0.92	0.93	0.92	1	1	1
Jogging	0.83	0.87	0.85	0.99	0.99	0.99	0.89	0.99	0.94	0.99	0.81	0.89	0.99	1	1
Free Running	0.98	0.82	0.9	0.99	1	0.99	0.98	0.89	0.93	0.81	0.92	0.86	1	0.99	1
Jumping Front & Back	0	0	0	0.88	0.89	0.88	0.89	0.74	0.81	0.82	0.68	0.74	0.97	1	0.98
Jumping Up & Down	0	0	0	0.81	0.86	0.83	0.78	0.9	0.83	0.81	0.81	0.81	0.92	0.91	0.91
Jumping Sideways	0.37	0.95	0.53	0.87	0.82	0.85	0.71	0.83	0.76	0.77	0.75	0.76	0.98	0.96	0.97
Jumping Jack	0.99	0.9	0.94	1	0.92	0.96	0.99	0.89	0.93	0.95	0.91	0.93	0.91	0.91	0.91
Arms Outstretch	0.96	0.89	0.93	1	0.99	1	1	0.99	0.99	0.85	0.89	0.87	1	0.99	0.99
Elbows Bent	0.84	0.98	0.9	1	0.94	0.97	0.98	0.94	0.96	0.9	0.94	0.92	0.98	1	0.99
Waist Rotation	0.81	0.74	0.77	0.89	1	0.94	0.94	1	0.97	0.81	0.8	0.81	0.96	1	0.98
Waist Bends Opposite Direction	0.93	0.9	0.92	1	0.99	0.99	1	1	1	0.97	0.89	0.93	0.99	0.98	0.98
Reach Heels Backwards	0.49	0.93	0.64	0.94	0.9	0.92	0.99	0.89	0.94	0.76	0.75	0.76	1	1	1
Lateral Bend to Left & Right	0.61	0.88	0.72	0.91	0.97	0.94	0.93	0.96	0.95	0.82	0.81	0.82	1	1	1
Lateral Bend Arms Up	0.86	0.34	0.49	0.96	0.91	0.94	0.96	0.93	0.94	0.82	0.74	0.78	0.95	1	0.97
Forward Stretching	0.58	0.61	0.6	0.98	1	0.99	1	0.98	0.99	0.83	0.83	0.83	1	0.94	0.97
Opposite Twist	0.29	0.5	0.37	0.97	0.92	0.94	0.99	0.91	0.95	0.64	0.54	0.58	1	1	1
Arms Lateral Elevation	0.66	0.88	0.75	0.92	0.88	0.9	0.92	0.88	0.9	0.91	0.88	0.9	0.99	0.98	0.98
Frontal Hand Claps	0.65	0.8	0.72	0.91	0.97	0.94	0.9	0.9	0.9	0.68	0.71	0.69	0.98	0.99	0.98
Arms Frontal Crossing	0.48	0.98	0.64	0.96	0.89	0.93	0.92	0.93	0.92	0.87	0.77	0.82	0.93	1	0.96
Shoulders High Amplitude Rotation	0	0	0	0.85	0.86	0.86	0.88	0.86	0.87	0.87	0.72	0.79	1	1	1
Shoulders Low Amplitude Rotation	0	0	0	0.92	0.93	0.93	0.86	0.89	0.87	0.8	0.72	0.76	1	1	1
Arms Inner Rotation	0	0	0	0.95	0.99	0.97	0.99	0.93	0.96	0.85	0.88	0.86	0.96	0.98	0.97
Knees to Beast	0.48	0.66	0.55	1	0.93	0.96	1	0.93	0.96	0.81	0.8	0.8	1	1	1
Heels to Backside	0.22	0.21	0.22	0.96	0.98	0.97	1	0.94	0.97	0.76	0.74	0.75	1	0.99	1
Knees Bending Forward	0.8	0.85	0.83	0.96	1	0.98	0.95	1	0.97	0.88	0.95	0.91	0.97	1	0.99
Rotation on Knees	0.47	0.4	0.43	1	1	1	1	0.98	0.99	0.84	0.92	0.88	1	1	1
Rowing	1	0.96	0.98	1	1	1	1	1	1	1	0.94	0.97	1	1	1
Elliptical Bike	0.74	0.73	0.73	0.99	0.96	0.97	0.96	0.98	0.97	0.96	0.7	0.81	1	1	1

TABLE 9. Comparison of classifiers' performance using HAR70+ dataset.

‘	DT			RF			AdaBoost			GBDT			AHDT		
Activity	P	R	F1	P	R	F1	P	R	F1	P	R	F1	hP	hR	hF1
Free Walking	0.98	0.99	0.98	0.98	0.99	0.99	0.98	0.98	0.98	0.98	0.96	0.97	1	1	1
shuffling	0.28	0.18	0.22	0.26	0.11	0.16	0.15	0.15	0.15	0.15	0.22	0.18	1	0.72	0.84
Ascending Stairs	0.37	0.49	0.42	0.57	0.49	0.53	0.75	0.44	0.55	0.01	0.09	0.02	0.87	0.73	0.79
Descending Stairs	0.32	0.29	0.31	0.88	0.27	0.42	1	0.1	0.18	0.02	0.12	0.04	0.9	0.71	0.79
standing	0.98	0.98	0.98	0.98	0.99	0.99	0.98	0.99	0.99	0.99	0.97	0.98	1	1	1
Sitting	0.94	0.94	0.94	0.99	0.93	0.96	0.98	0.94	0.96	0.99	0.93	0.96	1	1	1
lying	0.86	0.86	0.86	0.87	0.98	0.92	0.88	0.96	0.92	0.87	0.97	0.91	1	1	1

TABLE 10. CPU processing time taken by each algorithm in generating the classification model using various datasets.

	Classifier				
Dataset	DT	AHDT	RF	AdaBoost	GBDT
WISDM	97.3	154.83	1897.91	5401.59	21512.17
HARTH	482.06	813.08	5281.16	30273.06	147268.7
MHEALTH	22.03	41.5	74.81	911.39	9475.09
PAMAP2	57.14	103.02	305.3	2991.08	17108.16
DaLiAc	71.16	194.17	264.52	4037.83	16961.2
REALDISP	111.75	253.73	167.13	6356.63	30756.77
HAR70+	3057.55	37678.12	189619.79	886248.17	1671.87

V. CONCLUSION AND FUTURE WORK

This paper introduced an Adaptive Hierarchical Decision Tree (AHDT) HAR system that is more appropriate for learning ambiguous activities and more explicit to interpret with superior predictive performance. The fundamental contributions of the proposed system include 1) the hierarchical global classifier design and implementation based on the decision tree framework and 2) the introduction of elevated higher-level orientation (quasi-stationary inclination) features of the IMU sensor with adaptive segment processing. The proposed HAR system has been tested and evaluated on six publicly available data sets containing about 50 different activities of all levels of complexity. The experimental results offered exciting insights, showing that the proposed method produced favourable results in the context of HAR using the prior knowledge of the similarities between the activities. Results demonstrated that the proposed hierarchical tree classification improves accuracy, especially for complex ambiguity activities. These activities confuse enhanced tree-based methods Random Forest, Ada-boosting and Gradient boosting Decision tree. One of the advantages of the AHDT system is that it focuses more on extracting the most manageable, meaningful features instead of developing overly complicated, hard-to-interpret features.

Compared to other popular tree-based flat classification methods, our system stands out for its comprehensibility, computational efficiency, and interpretability. Unlike deep learning networks and tree-boosting methods, our proposed system can learn and generalize effectively even with a limited dataset. Additionally, the proposed AHDT system is less complex and more straightforward to modify than other tree-based and DL models. It can be easily visualized, and its white-box model provides clear explanations for the reasoning behind its predictions, which can be easily understood using Boolean logic. Moreover, the AHDT system can handle multi-level hierarchical classification with logarithmic complexity, making it efficient and cost-effective.

However, increasing the activity hierarchy depth may result in a complex tree-based model. Also, if certain classes are overrepresented at the leaf nodes of the hierarchy, it could result in biased trees. Therefore, it is important to balance the data samples at each node of the hierarchy to address this issue, which may result in added complexity.

A hierarchical representation of the sensor data collected from different body parts would be studied as a future work direction to present a more profound, detailed description of human activities. Also, examining the effect of hierarchical feature extraction integrated with hierarchical classification and measuring the real-time classification system performance are important future directions to prove the suitability of the proposed system for emerging health applications like assisted living.

REFERENCES

[1] Statistics Canada. (Nov. 2022). *Older Adults and Population Aging Statistics*. Accessed: Apr. 4, 2024. [Online]. Available: http://www.statcan.gc.ca/en/subjectsstart/olderadultsandpopulationaging

[2] P. Zimonjic. (Nov. 2022). *Canada's Working-Age Population is Older Than Ever, StatsCan Says*. Accessed: Apr. 4, 2024. [Online]. Available: https://www.cbc.ca/news/politics/canada-working-age-poupulation-census-1.6432398

[3] C. Aviles-Cruz, E. Rodriguez-Martinez, J. Villegas-Cortez, and A. Ferreyra-Ramirez, "Granger-causality: An efficient single user movement recognition using a smartphone accelerometer sensor," *Pattern Recognit. Lett.*, vol. 125, pp. 576–583, Jul. 2019.

[4] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, "A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1384–1393, Apr. 2019.

[5] J. A. Muñoz-Cristóbal, M. J. Rodríguez-Triana, V. Gallego-Lema, H. F. Arribas-Cubero, J. I. Asensio-Pérez, and A. Martínez-Monés, "Monitoring for awareness and reflection in ubiquitous learning environments," *Int. J. Human–Comput. Interact.*, vol. 34, no. 2, pp. 146–165, Feb. 2018.

[6] S. K. Yadav, K. Tiwari, H. M. Pandey, and S. A. Akbar, "A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions," *Knowl.-Based Syst.*, vol. 223, Jul. 2021, Art. no. 106970.

[7] M. Straczkiewicz, P. James, and J.-P. Onnela, "A systematic review of smartphone-based human activity recognition methods for health research," *NPJ Digit. Med.*, vol. 4, no. 1, pp. 1–15, Oct. 2021.

[8] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining Knowl. Discovery*, vol. 22, nos. 1–2, pp. 31–72, Jan. 2011.

[9] Y. Dong, R. Zhou, C. Zhu, L. Cao, and X. Li, "Hierarchical activity recognition based on belief functions theory in body sensor networks," *IEEE Sensors J.*, vol. 22, no. 15, pp. 15211–15221, Aug. 2022.

[10] A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano, "Trends in human activity recognition using smartphones," *J. Reliable Intell. Environ.*, vol. 7, no. 3, pp. 189–213, Sep. 2021.

[11] J. R. Quinlan, "Improved use of continuous attributes in C4.5," 1996, *arXiv:cs/9603103*.

[12] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, Apr. 2014.

[13] J. Zhang, H. Liang, and X. Wang, "Combined spectrogram structure and intensity for human activity recognition using modified DTW," in *Proc. IET International Radar Conference (IET IRC)*, Nov. 2020, pp. 1380–1385.

[14] B. Fu, N. Damer, F. Kirchbuchner, and A. Kuijper, "Sensing technology for human activity recognition: A comprehensive survey," *IEEE Access*, vol. 8, pp. 83791–83820, 2020.

[15] A. Wang, G. Chen, J. Yang, S. Zhao, and C.-Y. Chang, "A comparative study on human activity recognition using inertial sensors in a smartphone," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4566–4578, Jun. 2016.

[16] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. Int. Workshop Ambient Assist. Living*. Cham, Switzerland: Springer, 2012, pp. 216–223.

[17] C. A. Ronao and S.-B. Cho, "Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models," in *Proc. 10th Int. Conf. Natural Comput. (ICNC)*, Aug. 2014, pp. 681–686.

[18] H. Nematallah, S. Rajan, and A.-M. Cretu, "Logistic model tree for human activity recognition using smartphone-based inertial sensors," in *Proc. IEEE SENSORS*, Oct. 2019, pp. 1–4.

[19] F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi, "Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey," *IEEE Access*, vol. 8, pp. 210816–210836, 2020.

[20] N. Gupta, S. K. Gupta, R. K. Pathak, V. Jain, P. Rashidi, and J. S. Suri, "Human activity recognition in artificial intelligence framework: A narrative review," *Artif. Intell. Rev.*, vol. 55, no. 6, pp. 4755–4808, Aug. 2022.

[21] M. Inoue, S. Inoue, and T. Nishida, "Deep recurrent neural network for mobile human activity recognition with high throughput," *Artif. Life Robot.*, vol. 23, no. 2, pp. 173–185, Jun. 2018.

[22] C. Liu, J. Ying, F. Han, and M. Ruan, "Abnormal human activity recognition using Bayes classifier and convolutional neural network," in *Proc. IEEE 3rd Int. Conf. Signal Image Process. (ICSIP)*, Jul. 2018, pp. 33–37.

[23] R. Xi, M. Hou, M. Fu, H. Qu, and D. Liu, "Deep dilated convolution on multimodality time series for human activity recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[24] W. Xu, Y. Pang, Y. Yang, and Y. Liu, "Human activity recognition based on convolutional neural network," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 165–170.

[25] S. Ha and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 381–388.

[26] T. Sztyler, H. Stuckenschmidt, and W. Petrich, "Position-aware activity recognition with wearable devices," *Pervas. Mobile Comput.*, vol. 38, pp. 281–295, Jul. 2017.

[27] B. Bruno, F. Mastrogiovanni, and A. Sgorbissa, "Wearable inertial sensors: Applications, challenges, and public test benches," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 116–124, Sep. 2015.

[28] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mHealthdroid: A novel framework for agile development of mobile health applications," in *Proc. Int. Workshop Ambient Assist. Living*. Belfast, U.K.: Springer, 2014, pp. 91–98.

[29] H. Nematallah and S. Rajan, "Comparative study of time series-based human activity recognition using convolutional neural networks," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2020, pp. 1–6.

[30] B. Weigel, K. Loar, A. Colón, and R. Wright, "What did our model just learn? Hard lessons in applying deep learning to human factors data," in *Proc. Int. Conf. Appl. Human Factors Ergonom.* USA: Springer, 2021, pp. 52–60.

[31] Y. Kwon, K. Kang, and C. Bae, "Unsupervised learning for human activity recognition using smartphone sensors," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6067–6074, Oct. 2014.

[32] Y. Li, D. Shi, B. Ding, and D. Liu, "Unsupervised feature learning for human activity recognition using smartphone sensors," in *Mining Intelligence and Knowledge Exploration*. Cork, Ireland: Springer, 2014, pp. 99–107.

[33] J. Lu, X. Zheng, M. Sheng, J. Jin, and S. Yu, "Efficient human activity recognition using a single wearable sensor," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11137–11146, Nov. 2020.

[34] T. Huynh-The, C.-H. Hua, and D.-S. Kim, "Visualizing inertial data for wearable sensor based daily life activity recognition using convolutional neural network," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 2478–2481.

[35] T. Hur, J. Bang, T. Huynh-The, J. Lee, J.-I. Kim, and S. Lee, "Iss2Image: A novel signal-encoding technique for CNN-based human activity recognition," *Sensors*, vol. 18, no. 11, p. 3910, Nov. 2018.

[36] T. Huynh-The, C.-H. Hua, N. A. Tu, and D.-S. Kim, "Physical activity recognition with statistical-deep fusion model using multiple sensory data for smart health," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1533–1543, Feb. 2021.

[37] I. Debache, L. Jeantet, D. Chevallier, A. Bergouignan, and C. Sueur, "A lean and performant hierarchical model for human activity recognition using body-mounted sensors," *Sensors*, vol. 20, no. 11, p. 3090, May 2020.

[38] Y. Tang, Q. Teng, L. Zhang, F. Min, and J. He, "Layer-wise training convolutional neural networks with smaller filters for human activity recognition using wearable sensors," *IEEE Sensors J.*, vol. 21, no. 1, pp. 581–592, Jan. 2021.

[39] T. Lv, X. Wang, L. Jin, Y. Xiao, and M. Song, "Margin-based deep learning networks for human activity recognition," *Sensors*, vol. 20, no. 7, p. 1871, Mar. 2020.

[40] W. Gao, L. Zhang, Q. Teng, J. He, and H. Wu, "DanHAR: Dual attention network for multimodal human activity recognition using wearable sensors," *Appl. Soft Comput.*, vol. 111, Nov. 2021, Art. no. 107728.

[41] S. P. Singh, M. K. Sharma, A. Lay-Ekuakille, D. Gangwar, and S. Gupta, "Deep ConvLSTM with self-attention for human activity decoding using wearable sensors," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8575–8582, Mar. 2021.

[42] Y. Wang, X. Wang, H. Yang, Y. Geng, H. Yu, G. Zheng, and L. Liao, "MhaGNN: A novel framework for wearable sensor-based human activity recognition combining multi-head attention and graph neural networks," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–14, 2023.

[43] S. Khan, S. H. Noorani, A. Arsalan, A. Mahmood, U. Rauf, and Z. Ali, "Classification of human physical activities and postures during everyday life," in *Proc. 18th Int. Conf. Emerg. Technol. (ICET)*, Nov. 2023, pp. 98–103.

[44] F. Hajjej, M. Javeed, A. Ksibi, M. Alarfaj, K. Alnowaiser, A. Jalal, N. Alsufyani, M. Shorfuzzaman, and J. Park, "Deep human motion detection and multi-features analysis for smart healthcare learning tools," *IEEE Access*, vol. 10, pp. 116527–116539, 2022.

[45] M. Awais, L. Chiari, E. A. F. Ihlen, J. L. Helbostad, and L. Palmerini, "Physical activity classification for elderly people in free-living conditions," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 1, pp. 197–207, Jan. 2019.

[46] Y. Kaya and E. K. Topuz, "Human activity recognition from multiple sensors data using deep CNNs," *Multimedia Tools Appl.*, vol. 83, no. 4, pp. 10815–10838, Jan. 2024.

[47] A. Dahou, M. A. A. Al-Qaness, M. A. Elaziz, and A. M. Helmi, "MLCNNwav: Multilevel convolutional neural network with wavelet transformations for sensor-based human activity recognition," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 820–828, 2024.

[48] R. A. Stein, P. A. Jaques, and J. F. Valiati, "An analysis of hierarchical text classification using word embeddings," *Inf. Sci.*, vol. 471, pp. 216–232, Jan. 2019.

[49] H. Zhao, P. Wang, Q. Hu, and P. Zhu, "Fuzzy rough set based feature selection for large-scale hierarchical classification," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 10, pp. 1891–1903, Oct. 2019.

[50] H. Wang, J. Zhao, J. Li, L. Tian, P. Tu, T. Cao, Y. An, K. Wang, and S. Li, "Wearable sensor-based human activity recognition using hybrid deep learning techniques," *Secur. Commun. Netw.*, vol. 2020, pp. 1–12, Jul. 2020.

[51] S. Scheurer, S. Tedesco, K. N. Brown, and B. O'Flynn, "Using domain knowledge for interpretable and competitive multi-class human activity recognition," *Sensors*, vol. 20, no. 4, p. 1208, Feb. 2020.

[52] A. Akbari, J. Wu, R. Grimsley, and R. Jafari, "Hierarchical signal segmentation and classification for accurate activity recognition," in *Proc. ACM Int. Joint Conf. Int. Symp. Pervasive Ubiquitous Comput. Wearable Comput.*, Oct. 2018, pp. 1596–1605.

[53] F. De Rango, M. Tropea, A. Serianni, and N. Cordeschi, "Fuzzy inference system design for promoting an eco-friendly driving style in IoV domain," *Veh. Commun.*, vol. 34, Apr. 2022, Art. no. 100415.

[54] M. Fazli, K. Kowsari, E. Gharavi, L. Barnes, and A. Doryab, "HHAR-Net: Hierarchical human activity recognition using neural networks," in *Proc. Int. Conf. Intell. Human Comput. Interact.* Berlin, Germany: Springer, 2021, pp. 48–58.

[55] A. Wang, G. Chen, X. Wu, L. Liu, N. An, and C.-Y. Chang, "Towards human activity recognition: A hierarchical feature selection framework," *Sensors*, vol. 18, no. 11, p. 3629, Oct. 2018.

[56] F. J. M. Shamrat, R. Ranjan, K. Md, A. Y. Hasib, and A. H. Siddique, "Performance evaluation among ID3, C4.5, and CART decision tree algorithms," in *Proc. Pervasive Comput. Social Netw. (ICPCSN)*, vol. 317, 2021, pp. 127–142.

[57] F. Esposito, D. Malerba, G. Semeraro, and J. Kay, "A comparative analysis of methods for pruning decision trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 476–491, May 1997.

[58] Y. Cao, Q.-G. Miao, J.-C. Liu, and L. Gao, "Advance and prospects of AdaBoost algorithm," *Acta Automatica Sinica*, vol. 39, no. 6, pp. 745–758, Mar. 2014.

[59] G. Louppe, "Understanding random forests: From theory to practice," 2014, *arXiv:1407.7502*.

[60] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.

[61] A. Logacjov, K. Bach, A. Kongsvold, H. B. Bårdstu, and P. J. Mork, "HARTH: A human activity recognition dataset for machine learning," *Sensors*, vol. 21, no. 23, p. 7853, Nov. 2021.

[62] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.

[63] H. Leutheuser, D. Schuldhaus, and B. M. Eskofier, "Hierarchical, multi-sensor based classification of daily life activities: Comparison with state-of-the-art algorithms using a benchmark dataset," *PLoS ONE*, vol. 8, no. 10, Oct. 2013, Art. no. e75196.

[64] O. Baños, M. Damas, H. Pomares, I. Rojas, M. A. Tóth, and O. Amft, "A benchmark dataset to evaluate sensor displacement in activity recognition," in *Proc. ACM Conf. Ubiquitous Comput.*, Sep. 2012, pp. 1026–1035.

[65] A. Ustad, A. Logacjov, S. Ø. Trollebø, P. Thingstad, B. Vereijken, K. Bach, and N. S. Maroni, "Validation of an activity type recognition model classifying daily physical behavior in older adults: The HAR70+ model," *Sensors*, vol. 23, no. 5, p. 2368, Feb. 2023.

HEBA NEMATALLAH (Graduate Student Member, IEEE) received the B.Sc. degree from the Faculty of Computer Science and Information Technology, Ain Shams University, Cairo, Egypt, and the Master of Applied Science degree in computer engineering from Queens University. She is currently pursuing the Ph.D. degree with Carleton University, ON, Canada. She also has research and development industry experience in natural language processing, speech processing, and vision-aided navigation systems. Her research interests include sensor fusion, activity recognition, and machine learning applications in healthcare.

SREERAMAN RAJAN (Senior Member, IEEE) received the B.E. degree in electronics and communications from Bharathiyar University, Coimbatore, India, in 1987, the M.Sc. degree in electrical engineering from Tulane University, New Orleans, LA, USA, in 1992, and the Ph.D. degree in electrical engineering from the University of New Brunswick, Fredericton, NB, Canada, in 2004. From 1986 to 1990, he was a Scientific Officer with the Reactor Control Division, Bhabha Atomic Research Center (BARC), Bombay, India, after undergoing intense training in nuclear science and engineering from its training school. At BARC, he developed systems for control, safety, and regulation of nuclear research and power reactors. From 1997 to 1998, he carried out research under a grant from Siemens Corporate Research, Princeton, NJ, USA. From 1999 to 2000, he was with JDS Uniphase, Ottawa, ON, Canada, where he worked on optical components and the development of signal processing algorithms for advanced fiber optic modules. From 2000 to 2003, he was with Ceyba Corporation, Ottawa, where he developed channel monitoring, dynamic equalization, and optical power control solutions for advanced ultra-long haul and long haul fibre optic communication systems. In 2004, he was with Biopeak Corporation, where he developed signal-processing algorithms for non-invasive medical devices. From December 2004 to June 2015, he was a Defense Scientist with the Defence Research and Development Canada, Ottawa. He joined Carleton University as a Tier 2 Canada Research Chair (Advanced Sensors Systems and Signal Processing) in its Department of Systems and Computer Engineering, in July 2015, and is currently a Full Professor. He was the Director of Ottawa-Carleton Institute for Biomedical Engineering (OCIBME), from 2020 to 2022. He was an Adjunct Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada (July 2010–June 2018) and is an Adjunct Professor with the Department of Electrical and Computer Engineering, Royal Military College, Kingston, ON, Canada, since July 2015. He is the holder of two patents and two disclosures of the invention. He is the author of more than 200 journal articles and conference papers. His research interests include signal and image processing, biomedical signal processing, pattern classification, and applied machine learning. He is currently the Chair of the IEEE Ottawa EMBS and AESS Chapters. He has been the North America Regional Director for the IEEE Consumer Technology Society, since 2021, and is a member-at-large in its Board of Governors. He has served IEEE Canada as it's a Board Member (2010–2018). He was awarded the IEEE MGA Achievement Award, in 2012, and recognized for his IEEE contributions with Queen Elizabeth II Diamond Jubilee Medal, in 2012, IEEE Canada recognized his outstanding service through the 2016 W. S. Read Outstanding Service Award. He has been involved in organizing several successful IEEE conferences and has been a reviewer of several IEEE journals and conferences.