August 2018
Master's Degree Thesis

Wearable Sensor-Based Human Activity Recognition with Deep Recurrent Neural Networks

Graduate School of Chosun University

Department of Information and Communication Engineering

Abdulmajid Murad

Wearable Sensor-Based Human Activity Recognition with Deep Recurrent Neural Networks

깊은 순환 신경망을 이용한 웨어러블 센서 기반 인간 행동 인식

August 24, 2018

Graduate School of Chosun University

Department of Information and Communication Engineering

Abdulmajid Murad

Wearable Sensor-Based Human Activity Recognition with Deep Recurrent Neural Networks

Advisor: Prof. Jae-Young Pyun

A thesis submitted in partial fulfillment of the requirements for a master's degree

April 2018

Graduate School of Chosun University

Department of Information and Communication Engineering

Abdulmajid Murad

Acknowledgement

First, I would like to express my sincere gratitude to my advisor Prof. Jae-Young Pyun for his invaluable support, encouragement, supervision, personal guidance, and useful suggestions throughout the course of my research work. I have been lucky to have a supervisor who cared about my work and who responded to my questions and queries so promptly.

Beside my advisor, I would like to thank the committee members, Prof. Young-Sik Kim and Prof. Goo-Rak Kwon, for their encouragement and insightful comments.

I have great pleasure in acknowledging my fellow lab mates of Wireless and Mobile Computing Systems Lab, for the stimulating discussions and for all the fun we have had in the last two years.

Finally, I must express my profound gratitude to the biggest source of my strength, my family. This accomplishment would not been possible without their love, support, and continuous encouragement throughout my years of study. I will be grateful forever for your love.

압둘마지드 무라드의 석사학위논문을 인준함

위원장 조선대학교 교수 권구락 (인)

위 원 조선대학교 교수 김영식 (인)

위 원 조선대학교 교수 변재영 (인)

2018년 5월

조선대학교 대학원

Table of Contents

List of Figures

List of Tables

Acronyms

ADL	Activities of Daily Living
ANN	Artificial Neural Networks
BN	Bayesian Network
CNN	Convolutional Neural Networks
DBN	Deep Believe Networks
DRNN	Deep Recurrent Neural Networks
ELM	Extreme Learning Machine
FOG	Freezing of Gait
HAD	Human Activity Dataset
HAR	Human Activity Recognition
HMM	Hidden Markov Models
KNN	K-Nearest Neighbors
LS-SVM	Least Square Support Vector Machines
LSTM	Long Short-Term Memory
NB	Naive Bayes
PCA	Principle Component Analysis
PD	Parkinson's Disease
RBM	Restricted Boltzmann Machine
RNN	Recurrent Neural Networks
SAE	Stacked Autoencoders
SVM	Support Vector Machines

Abstract

Wearable Sensor-Based Human Activity Recognition with Deep Recurrent Neural Networks

Abdulmajid Murad
Advisor: Prof. Jae-Young Pyun
Department of Information and Communication Engineering
Graduate School of Chosun University

Adopting deep learning methods for human activity recognition has been effective in extracting discriminative features from raw input sequences acquired from body-worn sensors. Although human movements are encoded in a sequence of successive samples in time, typical machine learning methods perform recognition tasks without exploiting the temporal correlations between input data samples. Convolutional neural networks (CNN) address this issue by using convolutions across a one-dimensional temporal sequence to capture dependencies among input data. However, the size of convolutional kernels restricts the captured range of dependencies between data samples. As a result, typical models are unadaptable to a wide range of activity-recognition configurations and require fixed-length input windows. In this thesis, we propose the use of deep recurrent neural networks (DRNN) for building recognition models that are capable of capturing long-

range dependencies in variable-length input sequences. We present unidirectional, bidirectional, and cascaded architectures based on long short-term memory (LSTM) DRNN and evaluate their effectiveness on miscellaneous benchmark datasets. Experimental results show that our proposed models outperform methods employing conventional machine learning, such as support vector machines (SVM) and k-nearest neighbors (KNN). Additionally, the proposed models yield better performance than other deep learning techniques, such as deep believe networks (DBN) and CNN.

한글요약

깊은 순환 신경망을 이용한 웨어러블 센서 기반 인간 행동 인식

압둘마지드 무라드
지도교수 :변재영
조선대학교대학원,
정보통신공학과

인간 행동 인 식을 위하여 심화 학습 방법들을 선택하는 것은 신체에 착용한 센서로부터 획득한 원시 입력 시퀀스로부터 차별된 특징을 추출하는데 효과적이다. 인간의 움직임은 시간경과에 따라 연속적인 샘플들로 인코딩되지만, 일반적인 기계 학습 방법들은 시간적 입력 데이터 샘플들 간의 상관관계를 활용하지 않고 인식 작업들을 수행한다. 컨볼루션 신경망 (Convolutional neural networks: CNN)은 입력 데이터간의 종속성을 획득하기 위해 1 차원 시간적 시퀀스에 따른 컨볼루션을 사용하여 이문제를 해결하지만, 컨볼루션 커널의 크기는 획득한 데이터 샘플간의 종속성 범위를 제한한다. 결과적으로,

일반적인 모델은 광범위한 행동인식 구성에 적합하지 않으며 고정된 길이의 입력창을 필요로한다. 본 연구에서, 저는 가변적인 길이를 갖는 입력 시퀀스에서 긴 범위의 종속성을 획득할수 있는 인식모델 설계를 위하여 깊은 순환 신경망 (Deep recurrent neural networks: DRNN)을 이용하는 것을 제안한다. 저는 단방향, 양방향 및 계단식 아키텍쳐를 기반으로하는 LSTM(Long short-term memory) DRNN 을 제안하고, 다양한 벤치마크 데이터셋들의 효율성을 평가한다. 실험결과는 제가 제안하는 모델들이 기존 SVM (Support Vector Machines)과 KNN (k-nearest neighbors)같은 기계학습 방법을 사용하는 것보다 성능이 우수함을 보여준다. 추가적으로, 제안하는 모델은 DBN (Deep Believe Neworks)와 CNN 같은 다른 심화 학습 기술보다 우수한 성능을 제공한다.

Chapter 1: Introduction

1.1 Motivation

Human activity recognition (HAR) has recently attracted increased attention from both researchers and industry with the goal of advancing ubiquitous computing and human computer interactions. It has many real-world applications, ranging from healthcare to personal fitness, gaming, tactical military applications, and indoor navigation. There are two major types of HAR: systems that use wearable sensors and systems that use external devices, such as cameras and wireless RF modules. In sensor-based HAR, wearable sensors are attached to a human body and the human activity is translated into specific sensor signal patterns that can be segmented and identified.

The application of deep learning for HAR has led to significant enhancements in recognition accuracy by overcoming many of the obstacles encountered by traditional machine learning methods. It provides a data-driven approach for learning efficient discriminative features from raw data, resulting in a hierarchy from low-level features to high-level abstractions. The strength of deep learning lies in its ability to automatically extract features in a task dependent manner. It avoids reliance on heuristic hand-crafted features and scales better for more complex behavior-recognition tasks.

The widespread use and availability of sensing technologies is generating an ever-growing amount of data, which along with enhanced computation power have contributed to more feasible applications of deep learning methods. These methods can be utilized to extract valuable contextual information from

physical activities in an unconstrained environment. Furthermore, many researchers have employed deep learning approaches to build HAR models in an end-to-end fashion, thereby achieving superior performance compared to previous conventional methods. This strategy has been effective in handling complex human activities and taking advantage of the proliferating data.

Recently, various deep learning and machine learning methods have been adopted for building activity recognition systems. These systems have considerably matured and performed with sufficient quality. However, it is always desirable to improve HAR systems more by exploring approaches not commonly tried before.

1.2 Objectives

The major objective of this research is to study the state-of-art methods used in HAR and identify potentials for improvements. We also aim to suggest and evaluate novel deep learning methods that can improve the overall accuracy and performance of HAR systems.

1.3 Contributions

In this thesis, we propose the use of long short-term memory (LSTM)-based deep recurrent neural networks (DRNN) to build HAR models. These models can classify activities mapped from variable-length input sequences. We develop architectures based on deep layers of unidirectional and bidirectional RNN, independently, as well as a cascaded architecture progressing from bidirectional to unidirectional RNN. These models are then tested on various benchmark datasets to validate their performance and generalizability for a

large range of activity recognition tasks. The major contributions of our work are as follows:

a) We demonstrate the effectiveness of using unidirectional and bidirectional DRNN for HAR tasks without any additional data preprocessing or merging with other deep learning methods.
b) We implement bidirectional DRNN cascaded architectures for HAR models. To the best of our knowledge, this the first work to do so.
c) We introduce models that are able to classify variable-length windows of human activities. This is accomplished by utilizing RNN's capacity to read variable-length sequences of input samples and merge the prediction for each sample into a single prediction for the entire window segment.

1.4 Thesis Layout

The reminder of this thesis is organized as follows. Chapter 2 provides background overview of HAR system, RNN, LSTM, and performance metrics commonly used in activity recognition. Chapter 3 discusses relevant previous work in the field of HAR. Chapter 4 describe the architectural structure of our proposed HAR system and the three DRNN models. Chapter 5 provide an overview of the experimental data that are used to train and evaluate our proposed models. Chapter 6 presents our experimental results where we apply the proposed models in different activity recognition domains and compare the models with other state-of-the-art HAR methods. Finally, chapter 7 concludes the thesis and presents summary of further research based on the presented work.

Chapter 2: Background

2.1 Human Activity Recognition System

HAR system is a system that focuses on the automatic recognition of physical activities performed by a subject. The system aims to provide accurate and valuable contextual information on people activities and behaviors. There are mainly two types of HAR: video-based and Sensor-based. Video-based HAR, an active research area in computer vision, analyzes image sequences or videos containing human motions for gestures and activities recognition. However, video-based HAR suffers from the constrained settings imposed by requiring a subject to perfume actions in front of a camera placed at a predefined position.

The focus of this thesis will be on Sensor-based HAR system. This system classifies human activities into a predefined set of classes based upon readings from wearable sensors placed on a monitored subject. It analyzes an input data, which are streams of time-series corresponding to frames of movement data, and predicts class labels for each frame.

The process of building sensor-based HAR pipeline is summarized in four steps as shown in figure 1. The first step is to collect data from multiple IMU sensors, and preprocess them with various signal-preprocessing techniques to remove noise and synchronize sensor measurements. Then segment the collected data into contiguous windows through a sliding-window approach. The segmentation step enables the system to look at longer segments of data rather than evaluating each data point separately. Thirdly, various time-domain or frequency-domain features are extracted from each window. These

features contains abstraction or reduced representation of the raw data and it is crucial to extract discriminative features that clearly separate between different activities. Lastly, an activity classifier is built using either machine learning or deep learning methods.

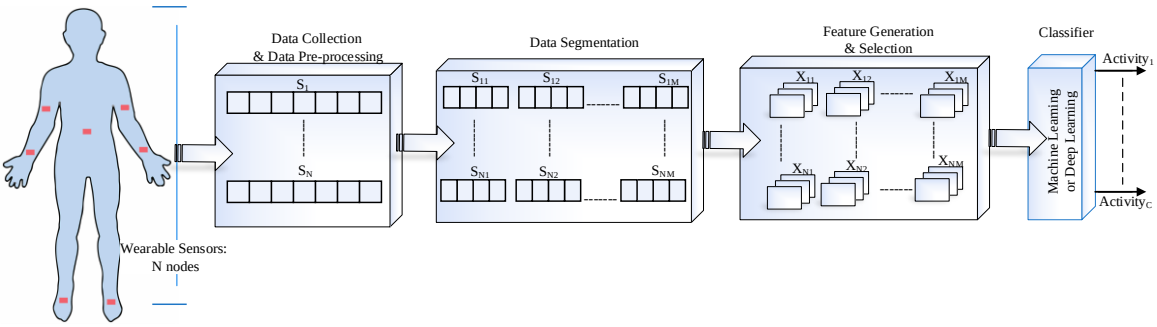


Figure1. Sensor-based HAR pipeline

Traditional machine learning methods such as Support Vector Machines (SVM) or K-Nearest Neighbor (KNN) were the standard for building HAR pipeline until recently. However, these methods rely on handcrafted feature extraction, which is a time-consuming heuristic, designed to a specific HAR task. This could hinder generalization to different HAR tasks. Fortunately, deep learning methods make it possible to perform automatic feature extraction by discovering patterns in the data. Recently, various deep learning methods have been adopted in HAR such as Artificial Neural Networks (ANN), Deep Belief Networks (DBN), Stacked Autoencoders (SAE), Convolutional Neural Networks (CNN), or hybrid of these methods. However, there have not been many research efforts in embracing Deep Recurrent Neural Networks (DRNN) for HAR, an issue addressed by this work.

2.2 Recurrent Neural Networks

In the field of deep learning, there is a growing interest in recurrent neural networks (RNN), which have been used for many sequence-modeling tasks. They have achieved promising performance enhancements in many technical applications, such as speech recognition [1], language modeling [2], video processing[3], and many other sequence labeling tasks [4]. The rationale behind their effectiveness for sequence-based tasks is their ability to exploit contextual information and learn the temporal dependencies in input sequences. RNN exploits temporal information by mapping the input into hidden states which are recurrently connected to itself [5].

2.2.1 Traditional RNN

An RNN is neural network architecture that contains cyclic connections, which enable it to learn the temporal dynamics of sequential data. A hidden layer in an RNN contains multiple nodes. As shown in Figure 2, each node has a function for generating the current hidden state h_t and output y_t by using its current input x_t and the previous hidden state h_{t-1} according to the following equations:

$$h_t = \mathcal{F}(W_h h_{t-1} + U_h x_t + b_h) \quad (1)$$

$$y_t = \mathcal{F}(W_y h_t + b_y), \quad (2)$$

where W_h, U_h, and W_y are the weight for the hidden-to-hidden recurrent connection, input-to-hidden connection, and hidden-to-output connection, respectively. b_h and b_y are bias terms for the hidden and output states, respectively. Additionally, there is an activation function $\mathcal{F}$ associated with

each node. This is an element-wise non-linearity function, commonly chosen from various existing functions, such as the sigmoid, hyperbolic tangent, or rectified linear unit (ReLU).

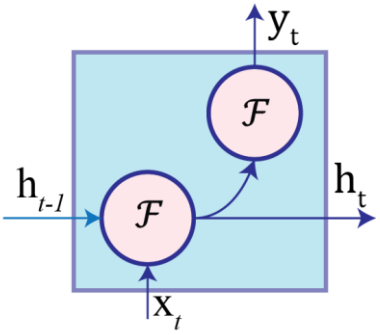


Figure 2. Schematic diagram of an RNN node where h_{t-1} is the previous hidden state, x_t is the current input sample, h_t is the current hidden state, y_t is the current output, and $\mathcal{F}$ is the activation function.

2.2.2 LSTM-based RNN

Training regular RNN can be challenging because of vanishing or exploding gradient problems that hinder the network's ability to backpropagate gradients through long-range temporal intervals [6]. This precludes modeling wide-range dependencies between input data for human activities when learning movements with long context windows. However, LSTM-based RNN can model temporal sequences and their wide-range dependencies by replacing the traditional nodes with memory cells that have internal and outer recurrence.

A memory cell contains more parameters and gate units, as shown in Figure 3. These gates control when to forget previous hidden states and when to update states with new information. The function of each cell component is as follows:

- Input gate i_t controls the flow of new information to the cell.
- Forget gate f_t determins when to forget content regarding the internal state.
- Output gate o_t controls which information flows to the output.
- Input modulation gate g_t is the main input to the cell.
- Internal state c_t handles cell internal recurrence.
- Hidden state h_t contains information from previously seen samples within the context window:

$$i_t = \sigma(b_i + U_i x_t + W_i h_{t-1}) \quad (3)$$

$$f_t = \sigma(b_f + U_f x_t + W_f x_{t-1}) \quad (4)$$

$$o_t = \sigma(b_o + U_o x_t + W_o h_{t-1}) \quad (5)$$

$$g_t = \sigma(b_g + U_g x_t + W_g h_{t-1}) \quad (6)$$

$$c_t = f_t c_{t-1} + g_t i_t \quad (7)$$

$$h_t = \tanh(c_t) o_t \quad (8)$$

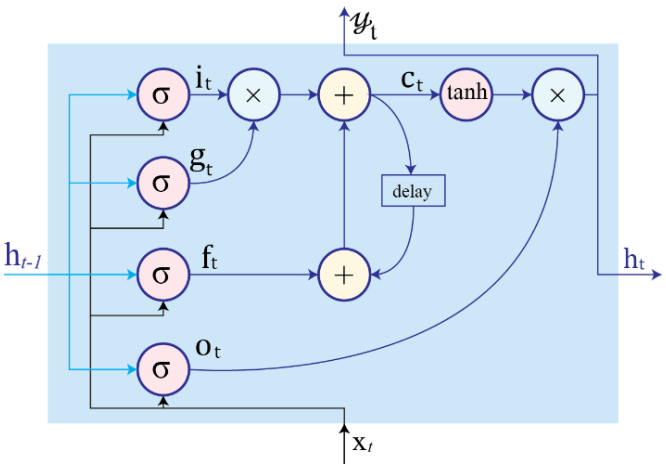


Figure 3. Schematic of LSTM cell structure with an internal recurrence c_t and an outer recurrence h_t. Cell gates are the input gate i_t, input modulation gate g_t, forget gate f_t, and output gate o_t. In contrast to an RNN node, the current output y_t is considered equal to current hidden state h_t.

The training process of LSTM-RNN is essentially focused on learning the parameters b, U, and W of the cell gates, as shown in Equations (3–6). However, RNN, and neural networks in general, suffer from overfitting training data. There are many techniques to deals with overfitting; the most common is dropout, which is a technique that randomly drop nodes and their connections during training [7]. Dropping out nodes reduces overfitting by preventing nodes from co-adapting too much. The choice of which nodes to drop is random and based upon dropout probability $\wp$.

2.3 Performance Metrics

To verify the performance of the proposed models, we employed four widely used evaluation metrics for multi-class classification [8]:

a) Precision: Measures the number of true samples out of those classified as positive. The overall precision is the average of the precisions for each class:

$$Per-class\ Precision_c = \frac{tp_c}{tp_c + fp_c} \tag{9}$$

$$Overall\ Precision = \frac{1}{\mathcal{C}}\left(\sum_{c=1}^{\mathcal{C}} \frac{tp_c}{tp_c + fp_c}\right) \tag{10}$$

where tp_c is the true positive rate of a class c, fp_c is the false positive rate, and $\mathcal{C}$ is the number of classes in the dataset.

b) Recall (Sensitivity): Measures the number correctly classified samples out of the total samples of a class. The overall recall is the average of the recalls for each class:

$$Per - class\ Recall_c = \frac{tp_c}{tp_c + fn_c} \tag{11}$$

$$Overall\ Recall\ = \frac{1}{\mathcal{C}}\left(\sum_{c=1}^{\mathcal{C}} \frac{tp_c}{tp_c + fn_c}\right) \tag{12}$$

where fn_c is the false negative rate of a class c.

c) Accuracy: Measures the proportion of correctly predicted labels over all predictions:

$$Overall\ Accuraccy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

where $TP = \sum_{c=1}^{\mathcal{C}} tp_c$ is the overall true positive rate for a classifier on all classes, $TN = \sum_{c=1}^{\mathcal{C}} tn_c$ is the overall true negative rate, $FP\ = \sum_{c=1}^{\mathcal{C}} fp_c$ is the overall false positive rate, and $FN = \sum_{c=1}^{\mathcal{C}} fn_c$ is the overall false negative rate.

d) F1-score: A weighted harmonic mean of precision and recall:

$$F1\ score = \sum_{c=1}^{\mathcal{C}} 2\left(\frac{n_c}{N}\right) \times \frac{precision_c \times recall_c}{precision_c + recall_c} \tag{14}$$

where n_c is the number of samples of a class c and $N = \sum_{c=1}^{\mathcal{C}} n_c$ is the total number of samples in a set with $\mathcal{C}$ classes. The F1-score is typically adopted for imbalanced datasets that have more samples of one class and less of another. Using accuracy as a performance metric in imbalanced datasets can be misleading, because any classifier can perform well by correctly classifying the majority class even if it wrongly classifies the minority class.

Chapter 3: Related Works

3.1 Traditional Approaches

HAR is a key research area the field of pattern recognition and has seen a tremendous progress in the last decade. The early work in HAR were in [9], but it was primitive and not practical. Decision trees, were utilized for HAR in [10],[11], in particular C4.5 and ID3 decision tree classifiers were used. In [12],[13], Bayesian Network (BN) and Naïve Bayes (NB) were adopted for activity recognition. In addition, K-nearest neighbors, which classify classes based upon the most similar class in the training set, were used in [10],[14]. The most common traditional method in HAR is SVM, which has been used extensively in the past decade such as in [15],[16],[17]. Fuzzy logic is another traditional approach used for HAR in [18],[19].

HAR has distinct research challenges that need to be addressed. In [20] a system was developed based on body-model derived features instead of low level signals, to overcome inter-person variability in systems trained for several people. These features are person-independent, thus robust to inter-person variability. Authors in [21] addressed interclass similarity by analyzing co-occurring activities and improved recognition rate by careful selection of individual features for each activity. In addition, authors in [22] used oversampling technique to overcome class imbalance challenge in pattern recognition in which samples of smaller class size are duplicated to equal the bigger class size. Furthermore, [23] addressed the challenge of collecting a large annotated training data by using semi-supervised techniques to leverage sparsely labeled data together with unlabeled data.

3.2 Deep Learning Approaches

3.2.1 Artificial Neural Networks

Recently, various deep learning methods have been adopted for building activity recognition systems. The simplest form of these methods is the traditional Artificial Neural Networks (ANN). The authors in [24] used deep layers of ANN, which are fed with extracted hand-engineered features from wrist-worn sensor, to construct a recognition system of users' complex activities. Similarly, in [25], ANN were used in combination with Principle Component Analysis (PCA) to build a classifier for HAR using mobile sensors. The PCA was used for dimensionality reduction of hand-engineered features and ANN were used as a classifier.

3.2.2 Deep Belief Networks

Deep Belief Networks (DBN) were the first deep learning method used in HAR. In [26], DBN were built by stacking multiple layers of restricted Boltzmann machine (RBM). Subsequent DBN-based models exploited the intrinsic temporal sequences in human activities by implementing hidden Markov models (HMM) above the RBM layers [27]. They performed an unsupervised pre-training step to generate intrinsic features and then used the available data labels to tune the model. However, HMM are limited by their numbers of possible hidden states, and could become impractical when modeling long-range dependencies in large context windows.

3.2.3 Stacked Autoencoders

Autoencoder is another deep learning approach that recently has been used in HAR. In [28], Staked Autoencoders (SAE) was used for constructing smartphones-based system to enhance the recognition accuracy and decrease recognition time. Furthermore, SAE was used in [29] to extract high level features form sensor data and integrate it with classifier training to build a jointly optimized framework for activity recognition in a Smart Home Environment.

3.2.4 Convolutional Neural Networks

The use of convolutional neural networks (CNN) for HAR was introduced in [30], but they used a shallow model and only a single accelerometer. Another model in [31] used deep CNN with only a single accelerometer. A multi-sensor recognition framework was developed in [32], where a deep CNN model for two accelerometers was proposed. A new multi-channel time series architecture of CNN was built in [33]. The architecture proposed in [34] was a compact model of shallow convolutional layers applied to the spectral domain of inertial signals. This model was optimized for low-power devices, but it reintroduced the extraction of handcrafted features by using a spectrogram of the input data.

The successful implementation of CNN for HAR is due to their capability for learning powerful and discriminative features, as well as utilizing convolutions across 1-D temporal sequence in order to capture local dependencies between nearby input samples. To capture local dependencies,

CNN use parameter sharing across time—applying the same convolutional kernel at each time segment—and local connectivity—neurons receiving inputs from small groups of input samples—between adjacent layers [35]. However, sharing parameters across time is insufficient for capturing all of the correlations between input samples. Additionally, local connectivity limits the output to a function of a small number of neighboring input samples.

3.2.5 Hybrid Models

CNN and RBM we combined in [36] into a hierarchical architecture to extract common bases of motion sensing data. In addition, CNN was combined with SAE in [37] , where CNN performs feature extraction and SAE, as a generative model, speeds up the training process.

3.2.6 Recurrent Neural Networks

Recurrent Neural Networks (RNN) have been used in many sequence modeling tasks such as speech recognition [1], language modeling [2], video processing[3], and many other sequence labeling tasks [4]. Furthermore, RNN showed promising results in video-based activity recognition [38]`[39]. However, the use of RNN in sensor-based HAR has not been investigated thoroughly. In [40], and integrated model of CNN and unidirectional RNN have been built, but it added complexity of combining multiple deep learning approaches in a single framework.

Chapter 4: Proposed HAR System

In this work, we propose the use of DRNN for HAR models in order exploit their internal memories for capturing the temporal dynamics of activity sequences. In contrast to [40], where CNN and RNN were used in a unified framework for activity recognition, our models are based only on DRNN, meaning we avoid the complexity of combining multiple deep learning approaches in a single framework. Additionally, by using only DRNN, our models are more flexible for classifying variable-length windows, in contrast to the fixed-length windows required by CNN. Bidirectional DRNN have been used in many domains, such as speech recognition and text-to-speech synthesis [1], [41], but, as far as we know, we are the first to use them in HAR models.

4.1 System Architecture

A schematic diagram of the proposed HAR system is presented in Figure 4. It performs direct end-to-end mapping from raw multi-modal sensor inputs to activity label classifications. It classifies the label of an activity performed during a specific time window. The input is a discrete sequence of equally spaced samples $(x_1, x_2, \dots, x_T)$, where each data point x_t is a vector of individual samples observed by the sensors at time t. These samples are segmented into windows of a maximum time index T then fed to an LSTM-based DRNN model.

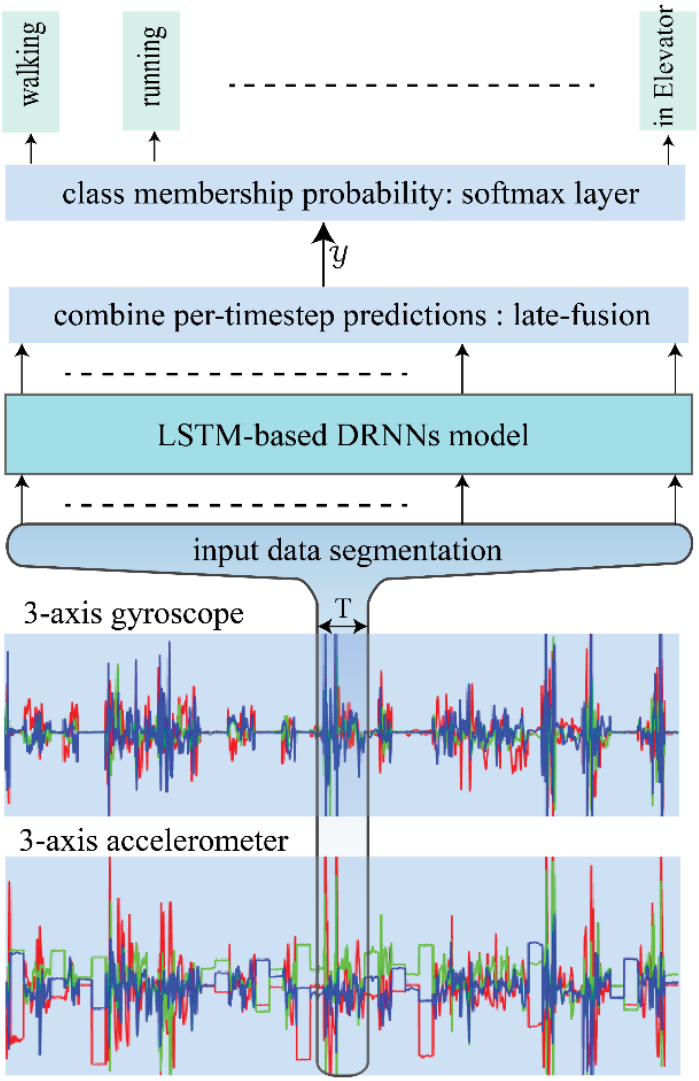


Figure 4. Proposed HAR system architecture. The inputs are raw signals obtained from multimodal-sensors, segmented into windows of length T and fed into LSTM-based DRNN model. The model outputs class prediction scores for each timestep, which are then merged via late-fusion and fed into the softmax layer to determine class membership probability.

The model outputs a sequence of scores representing activity label predictions in which there is a label prediction for each time step $(\mathcal{y}_1^L, \mathcal{y}_2^L, \ldots, \mathcal{y}_T^L)$, where $\mathcal{y}_t^L \in R^{\mathcal{C}}$ is a vector of scores representing the prediction for a given input sample x_t and $\mathcal{C}$ is the number of activity classes. There will a score for each time-step predicting the type of activity occurring at time t. The prediction for the entire window T is obtained by merging the individual scores into a single prediction. We have used late-fusion technique in which the classification decision from individual samples are combined for the overall prediction of a window. Using the "sum rule" in Equation (15) as the fusion scheme yields better results than other schemes, which is theoretically justified in [42]. We applied a softmax layer over $\mathcal{Y}$ to convert prediction scores into probabilities:

$$\mathcal{Y} = \frac{1}{T}\sum_{t=1}^{T} \mathcal{y}_t^L \tag{15}$$

In order to convert the prediction scores into probabilities, we used a softmax layer over $\mathcal{Y}$ to determine the estimated class membership probability p_k for a multi-class classification with $\mathcal{C}$ alternative classes:

$$p_k = \frac{\exp(W_k\,\mathcal{Y} + b_k)}{\sum_{k'=1}^{\mathcal{C}} \exp(W_{k'}\mathcal{Y} + b_{k'})} \quad (k = 1,2,\ldots\mathcal{C}) \tag{16}$$

where W_k and b_k are the weight and bias parameters for the softmax layer, respectively. The final prediction (labeling) for a segment window T is chosen based on the maximum posterior probability.

We have developed architectures for three DRNN models, which are as follows:

4.2 Unidirectional LSTM-Based DRNN Model

The first model is built using a unidirectional LSTM-based DRNN, as shown in Figure 5. Using sufficient number of DRNN layers can result in a very powerful model for transforming raw data into a more abstract representation, as well as for learning the temporal dependencies in time series data [1]. The input is a discrete sequence of equally spaced samples $(x_1,\ x_2, \dots, x_T)$, which are fed into the first layer at time t $(t = 1, 2, \dots, T)$.

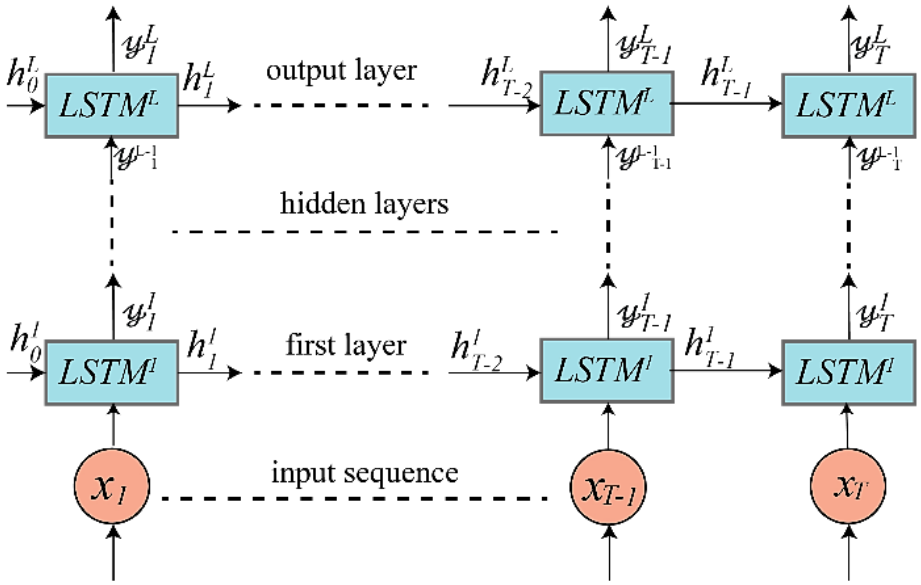


Figure 5. Unidirectional LSTM-based DRNN model consisting of an input layer, several hidden layers, and an output layer. The number of hidden layers is a hyperparameter that is tuned during training.

First, the hidden state h_0^{ℓ} and internal state c_0^{ℓ} of every layer ℓ are initialized to zeros. The first layer uses the input sample x_t at time t, previous hidden state h_{t-1}^1, and previous internal hidden state c_{t-1}^1 to generate the first layer output y_t^1 given its parameter θ^1 as follows:

$$y_t^1, h_t^1, c_t^1 = LSTM^1(c_{t-1}^1, h_{t-1}^1, x_t;\ \theta^1) \tag{17}$$

where θ^{ℓ} represents the parameters (b, U, W) of the LSTM cells for layer ℓ, as shown in Equations (3–6). Any layer ℓ in the upper layers uses the output of the lower layer $y_t^{\ell-1}$ as its input:

$$y_t^{\ell}, \mathrm{h}_t^{\ell}, c_t^{\ell} = \mathrm{LSTM}^{\ell}\left(\mathrm{c}_{t-1}^{\ell}, \mathrm{h}_{t-1}^{\ell}, y_t^{\ell-1};\ \theta^{\ell}\right). \tag{18}$$

The top layer L outputs $(y_1^L,\ y_2^L, \ldots, y_T^L)$, which is a sequence of scores representing the predictions at every time step in the window T.

4.3 Bidirectional LSTM-Based DRNN Model

The second model architecture is built by using a bidirectional LSTM-based DRNN, as shown in Figure 6. It includes two parallel LSTM tracks: forward and backward loops for exploiting context from the past and future of a specific time step in order to predict its label [43]. In the first layer, the forward track ($LSTM^{f1}$) reads the input window T from left to right, whereas the backward track ($LSTM^{b1}$) reads the input from right to left according to:

$$y_t^{f1}, h_t^{f1}, c_t^{f1} = LSTM^{f1}(c_{t-1}^{f1}, h_{t-1}^{f1}, x_t;\ W^{f1}) \tag{19}$$

$$y_t^{b1}, h_t^{b1}, c_t^{b1} = LSTM^{b1}\left(c_{t-1}^{b1}, h_{t-1}^{b1}, x_t;\ W^{b1}\right) \tag{20}$$

The top layer L outputs a sequence of scores at each time step for both forward LSTM $\left(y_1^{fL},\ y_2^{fL},\ldots,y_T^{fL}\right)$ and backward LSTM $\left(y_1^{bL},\ y_2^{bL},\ldots,y_T^{bL}\right)$. These scores are then combined into a single vector $\mathcal{Y} \in R^{\mathcal{C}}$ representing classes prediction for the window segment T. The late-fusion in this case will differ from that used in the unidirectional DRNN, Equation (15), because there are two outputs resulting from the forward and backward tracks, which are combined as follows:

$$\mathcal{Y} = \frac{1}{T}\sum_{t=1}^{T}\left(y_t^{fL} + y_t^{bL}\right) \tag{21}$$

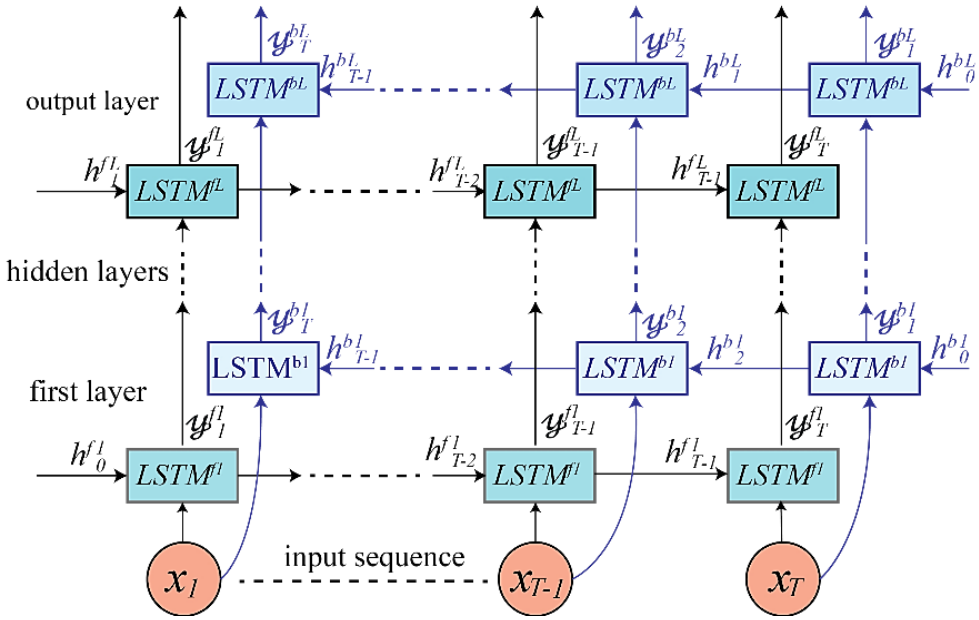


Figure 6. Bidirectional LSTM-based DRNN model consisting of an input layer, multiple hidden layers, and an output layer. Every layer has a forward $LSTM^{f\ell}$ and a backward $LSTM^{b\ell}$ track, and the number of hidden layers is a hyperparameter that is tuned during training.

4.4 Cascaded Bidirectional and Unidirectional LSTM-based DRNN Model

The third model architecture, shown in Figure 7, is motivated by [44]. It is a cascaded structure, in which the first layer is a bidirectional RNN and the upper layers are unidirectional. The first layer has a forward LSTM track $LSTM^{f1}$ that generates an output $\left(y_1^{f1},\ y_2^{f1}, \dots, y_T^{f1}\right)$ and a backward LSTM track $LSTM^{b1}$ that generates an output $\left(y_1^{b1},\ y_2^{b1}, \dots, y_T^{b1}\right)$, according to:

$$y_t^{f1}, h_t^{f1}, c_t^{f1} = LSTM^{f1}(c_{t-1}^{f1}, h_{t-1}^{f1}, x_t;\ W^{f1}) \tag{22}$$

$$y_t^{b1}, h_t^{b1}, c_t^{b1} = LSTM^{b1}\left(c_{t-1}^{b1}, h_{t-1}^{b1}, x_t;\ W^{b1}\right) \tag{23}$$

These two types of outputs are concatenated to form a new output $(y_1^1,\ y_2^1, \dots, y_T^1)$, which is fed into the second unidirectional layer

$$y_t^1 = y_t^{f1} + y_{T-t+1}^{b1} \tag{24}$$

The second layer only has one LSTM track (forward), which uses y_t^1 as its input according to:

$$y_t^2, \mathrm{h}_t^2, c_t^2 = \mathrm{LSTM}^2(\mathrm{c}_{t-1}^2, \mathrm{h}_{t-1}^2, y_t^1;\ \theta^2). \tag{25}$$

The upper layers uses the same functionality as the unidirectional model:

$$y_t^\ell, \mathrm{h}_t^\ell, c_t^\ell = \mathrm{LSTM}^\ell\left(\mathrm{c}_{t-1}^\ell, \mathrm{h}_{t-1}^\ell, y_t^{\ell-1};\ \theta^\ell\right). \tag{26}$$

The top layer L outputs $(y_1^L,\ y_2^L, \dots, y_T^L)$, which is a sequence of scores representing the predictions at every time step in the window T.

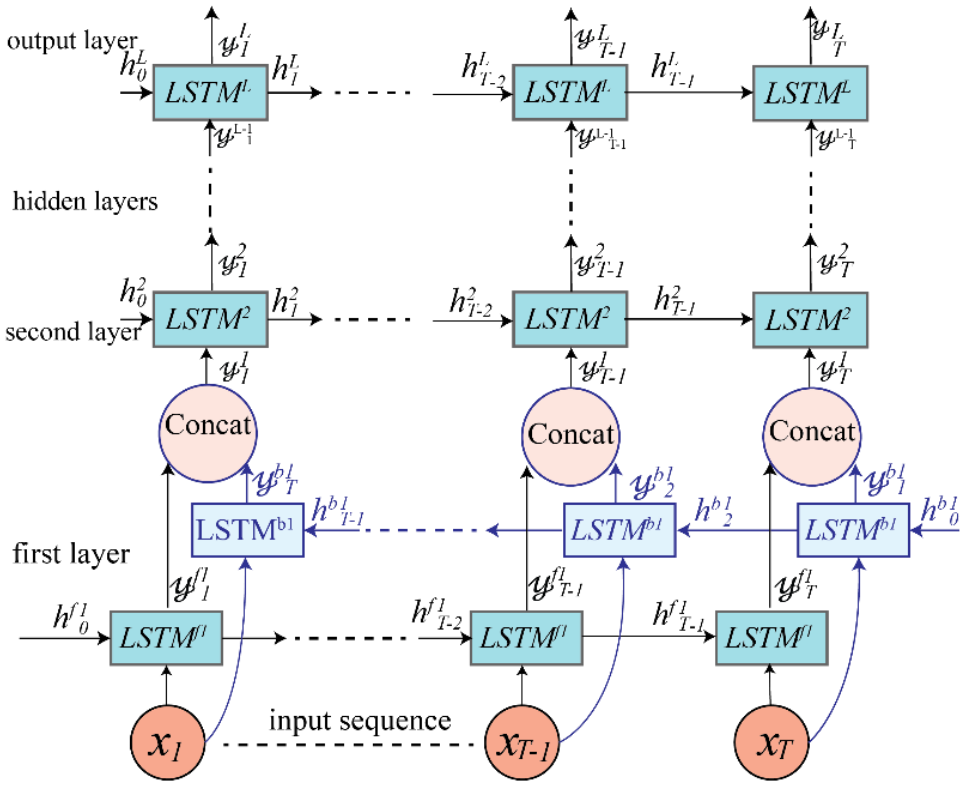


Figure 7. Cascaded bidirectional and unidirectional LSTM-based DRNN model. The first layer is bidirectional, whereas the upper layers are unidirectional. The number of hidden unidirectional layers is a hyperparameter that is tuned during training.

Chapter 5: Experimental Data

In order to train and evaluate the proposed models, we considered five public benchmark datasets for HAR. The datasets contain diverse movement data, captured by on-body sensors. They contain various activities performed in different environments and are used to validate the applicability and generalization of our models for a large variety of activity recognition tasks. Table 1 summarizes the experimental datasets and the following are brief descriptions of them:

a) UCI-HAD [45]: Dataset for activities of daily living (ADL) recorded by using a waist-mounted smartphone with an embedded 3-axis accelerometer, gyroscope, and magnetometer. All nine channels from the 3-axis sensors are used as inputs for our DRNN model at every time step. This dataset contains only six classes: walking, ascending stairs, descending stairs, sitting, standing, and laying.

b) USC-HAD [46]: Dataset collected by using a high performance IMU (3D accelerometer and gyroscope) sensor positioned on volunteers' front right hips. The dataset contains 12 basic human activities: walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping up, sitting, standing, sleeping, in elevator up, and in elevator down. We considered 11 classes by combining the last two activities into a single "in elevator" activity. The reason for this combination is that the model is unable to differentiate between the two classes using only a single IMU sensor. Additional barometer readings are

required to determine height changes in an elevator and discriminate between the two classes (up or down in elevator).

c) Opportunity [47]: Dataset comprised of ADL recorded in a sensor-rich environment. We consider only recordings from on-body sensors, which are seven IMUs and 12 3D-accelerometers placed on various body parts, as shown in Figure 8. There are 18 activity classes: opening and closing two types of doors, opening and closing three drawers at different heights, opening and closing a fridge, opening and closing a dishwasher, cleaning a table, drinking from a cup, toggling a switch, and a null-class for any non-relevant actions.

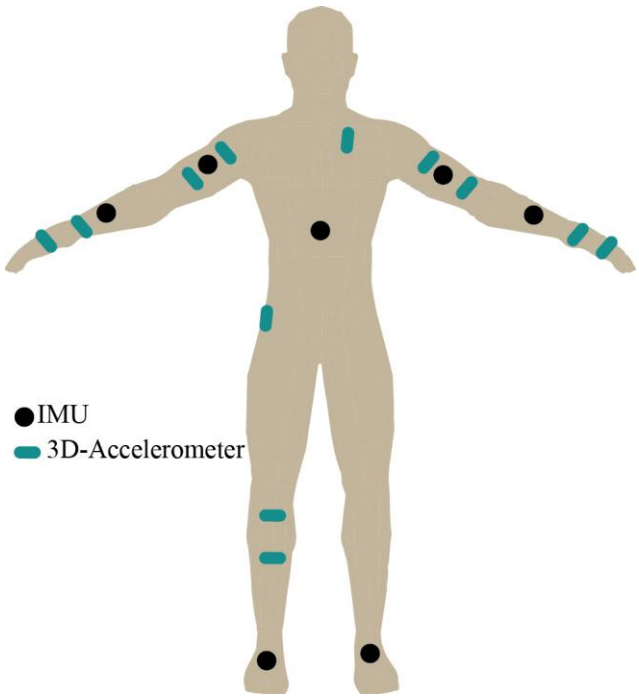


Figure 8. On-body sensors placement in Opportunity dataset

d) Daphnet FOG [48]: This dataset corresponds to a the medical application of activity recognition in gait analysis of patients with Parkinson's disease (PD). The dataset contains movement data from participants who suffer from a typical motor complication in PD and exhibit freezing of gait (FOG) symptoms. Three 3D-accelerometers were used to record the movement and they were attached to the shank, thigh, and lower back of the patients. Two classes (freeze and normal) were considered depending on whether or not the gait of a patient was frozen when the sample was recorded. We used this dataset to train our model to detect FOG episodes in PD patients and prove the suitability of our model for gait analysis using only wearable sensors.

e) Skoda [49]: Dataset containing activities of an employee in a car maintenance scenario. We consider recordings from a single 3D accelerometer, which is placed on the right hand of an employee. The dataset contains 11 activity classes: writing on a notepad, opening hood, closing hood, checking gaps on front door, opening left front door, closing left front door, closing both left doors, checking trunk gaps, opening and closing trunk, and a null-class for any non-relevant actions.

Table 1. Summary of human activity datasets used to evaluate the proposed deep learning models. Training window length indicates the number of samples in a window that we found to yield the best results for each dataset. Each dataset was divided into 80% for training and 20% for testing

Dataset	# of classes	Sensors	# of subjects	Sampling rate	Training window length	# of training examples	# of testing examples
UCI-HAD [45]	6	3D Acc., Gyro., and Magn. of a smartphone	30	50 Hz	128	11,988	2,997
USC-HAD [46]	12	3D Acc. & Gyro	14 (5 sessions)	100 Hz	128	44,000	11,000
Opportunity [47]	18	7 IMU sensors (3D ACC, Gyro & Mag.) & 12 Acc.	4 (5 sessions)	30 Hz	24	55,576	13,894
Daphnet FOG [48]	2	3 3D Acc.	10	64 Hz	32	57,012	14,253
Skoda [49]	11	3D Acc.	1 (19 sessions)	98 Hz	128	4411	1102

Chapter 6: Experimental Results and Discussion

6.1 Training Proposed Models

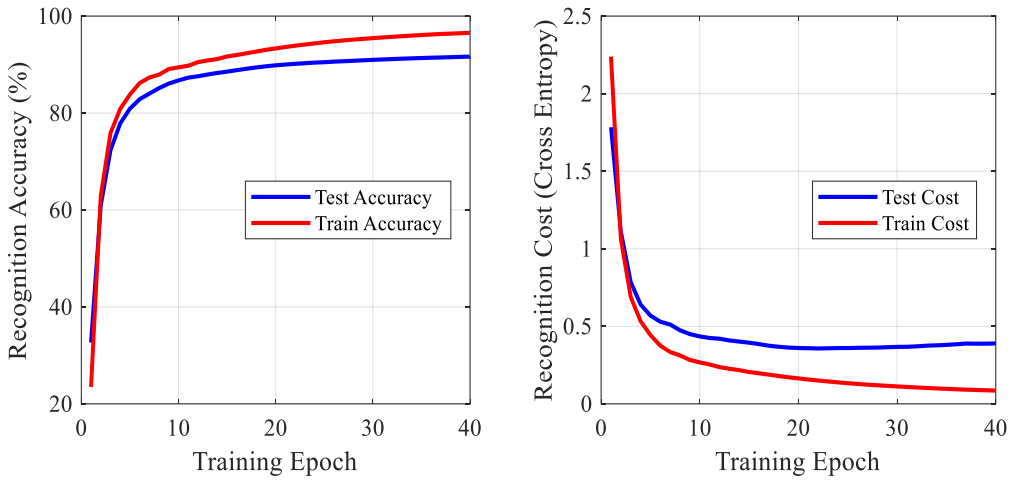
We trained our DRNN models on each dataset using 80% of the data for training and 20% for testing. The weights (parameters) of the models were initialized randomly and then updated to minimize a cost function $\mathcal{L}$. We used the mean cross entropy between the ground truth labels and the predicted output labels as the cost function. The ground truth labels are given in the datasets and indicate the true classes (labels) for the segmented windows. They are provided as a one-hot vector $\mathcal{O} \in R^{C}$ with a value o_k associated with each class k. The predicted label $\hat{\mathcal{O}} \in R^{C}$ contains the probability of every class p_k generated by our model:

$$\mathcal{L}(O, \hat{O}) = -\sum_{k=1}^{C} o_k \log p_k \tag{27}$$

We used an optimization algorithm called Adam that minimizes the cost function by backpropagating its gradient and updating model parameters [50]. Training was conducted on a GPU-based TensorFlow framework in order to utilize the parallel computation power of a GPU [51]. The dropout technique was used to avoid overfitting in our model [52]. Although dropout is typically applied to all nodes in a network, we followed the convention of applying dropout to the connections between layers (not on recurrent-connections or intra-cell connections). The probability of dropping a node during a training iteration is determined by the dropout probability $\mathcal{p}$, which is a hyperparameter tuned during training and represents the percentage of units to drop. Adopting dropout regularization technique led to a significant improvement in performance by preventing overfitting.

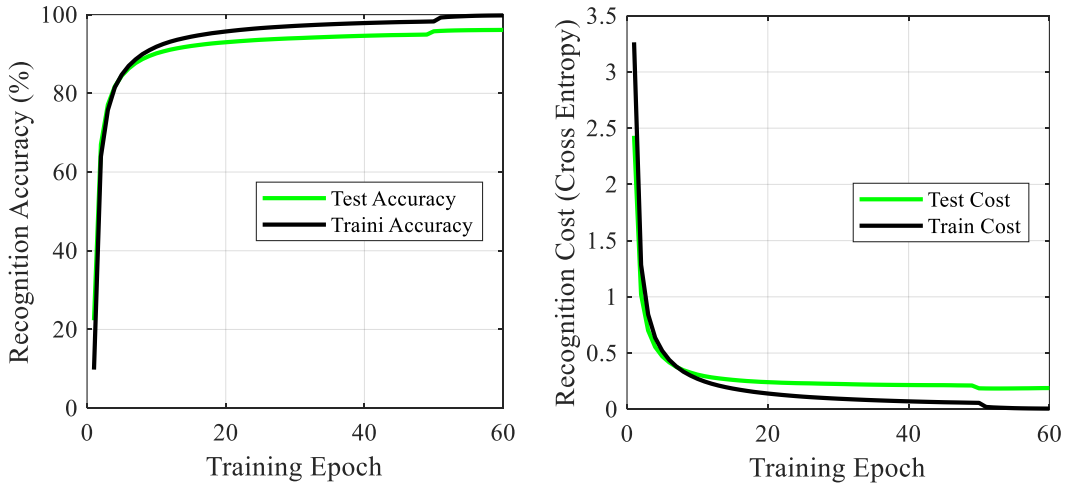
During training, the datasets were segmented with different window lengths, as outlined in Table 1. The optimal window length of a dataset depends on the sampling rate and the type of activities performed. We tested various lengths by “trial-and-error” method, then chose the window length that gave better performance results. Training was performed using the raw data without any further data preprocessing or intermediate intervention. The training and testing are generally performed using fixed-length windows, but the inputs of models may be using variable-length windows in the real-time data acquisition scenarios. In real-time application of HAR, data are captured over the course of time and the delay in DRNN is not fixed. Instead, the network can emit the corresponding label for a variable-length input segment. This is in contrast to other methods, such as CNN, in which the network must wait until a given fixed-length input segment is complete, before emitting the corresponding label.

Figure 9 to 13 present recognition accuracy and cross-entropy cost of training and testing processes for the proposed models using five different datasets. The gaps between training and testing accuracies, as well as the gaps between training and testing costs are small. This indicates that the dropout technique is very effective at forcing the model to generalize well and be resilient to overfitting.



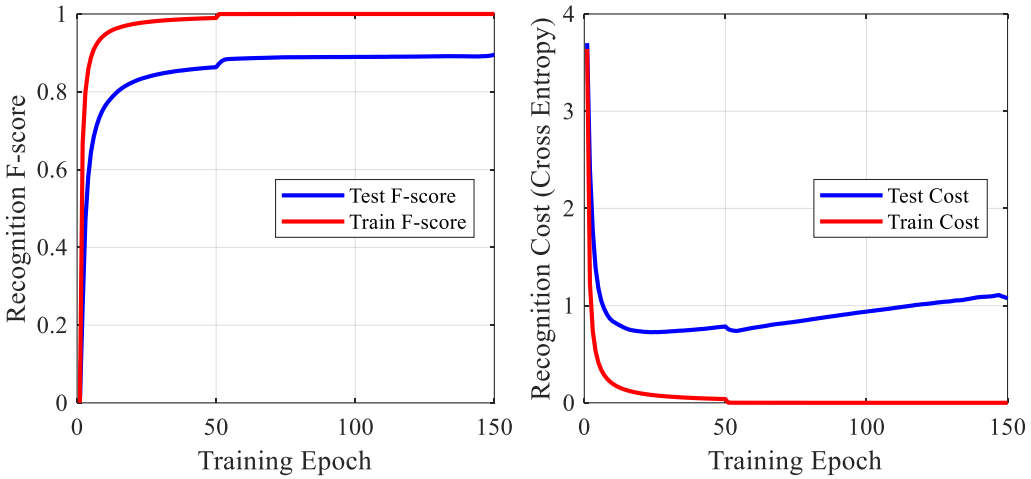
(**a**) (**b**)

Figure 9. Accuracy and cost of the unidirectional DRNN model for UCI-HAD dataset over training epochs: (**a**) training and testing accuracies; (**b**) cross-entropy costs between ground truth labels and predicted labels for both training and testing.



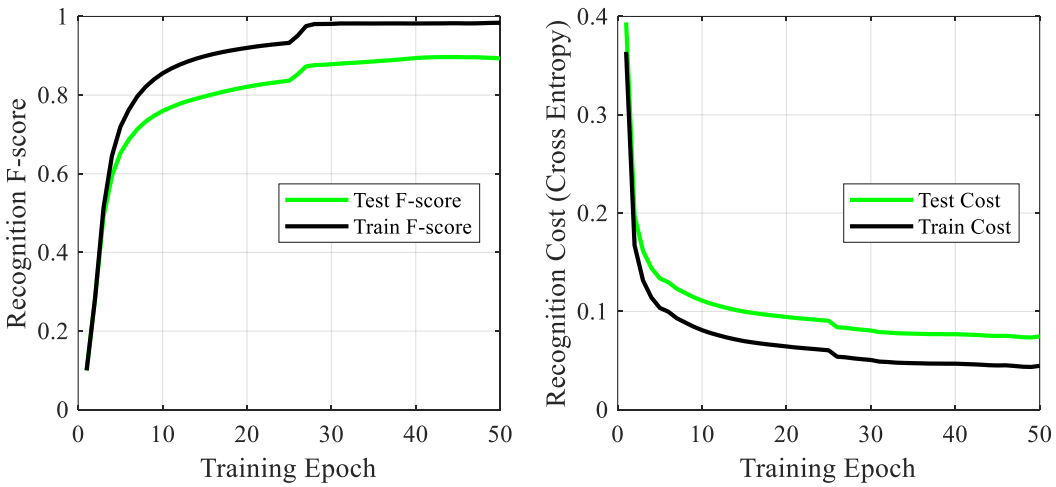
(**a**) (**b**)

Figure 10. Accuracy and cost of the unidirectional DRNN model for USC-HAD dataset over training epochs: (**a**) training and testing accuracies; (**b**) cross-entropy costs for both training and testing.



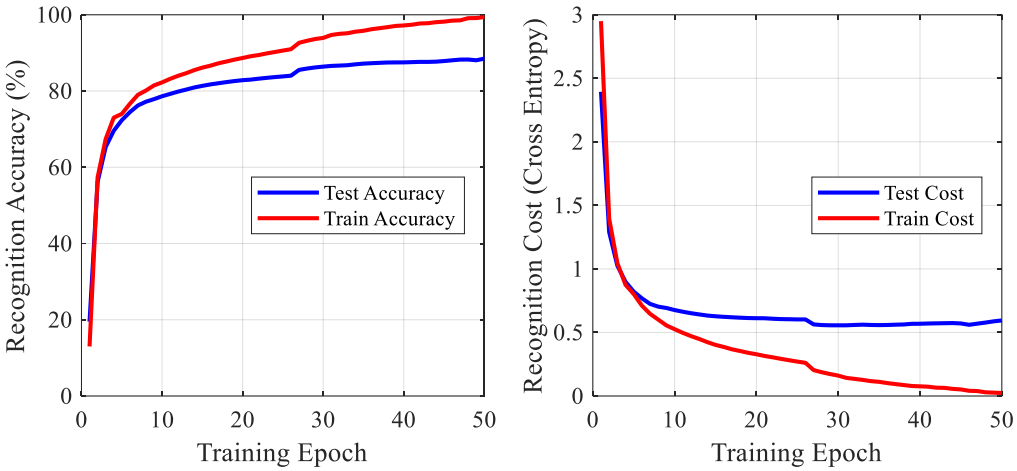
(**a**) (**b**)

Figure 11. F-score and cost of the bidirectional DRNN model for Opportunity dataset over training epochs: (**a**) training and testing F-scores; (**b**) cross-entropy costs between ground truth labels and predicted labels for both training and testing.



(**a**) (**b**)

Figure 12. F-score and cost of the cascaded DRNN model for Daphnet FOG dataset over training epochs: (**a**) training and testing F-scores; (**b**) cross-entropy costs between ground truth labels and predicted labels for both training and testing.



(**a**) (**b**)

Figure 13. Accuracy and cost of the cascaded DRNN model for Skoda dataset over training epochs: (**a**) training and testing accuracies; (**b**) cross-entropy costs between ground truth labels and predicted labels for both training and testing.

6.2 Performance Results

The performance results of our proposed models are presented in this section. The results are compared to other previously introduced methods, which are tested on the same datasets.

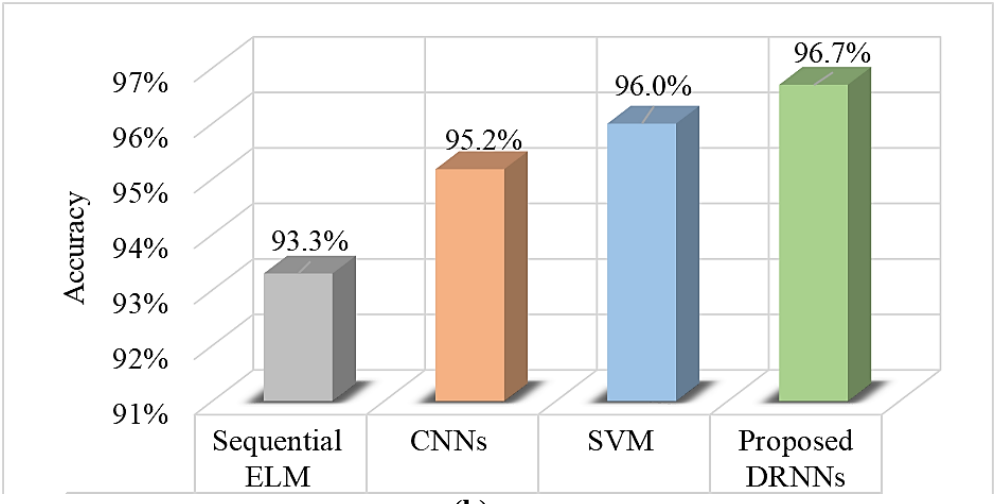
a) UCI-HAD

For the UCI-HAD dataset, we found that the unidirectional DRNN model with four layers yields best performance results in terms of per-class precision and recall, as shown in Figure 14a. The overall classification accuracy is 96.7%, outperforming other methods, such as CNN [53], support vector machines (SVM) [45], and sequential extreme learning machine (ELM) [54]. Figure 14b

presents a chart of the observed accuracy from our model in comparison with the accuracies achieved by other methods.

	Walking	W. Upstairs	W. Downstairs	Sitting	Standing	Laying	**Recall (%)**
Walking	510	5	7	0	0	0	98
W. Upstairs	6	462	13	0	0	0	96
W. Downstairs	1	4	426	0	0	0	99
Sitting	0	3	0	446	36	6	91
Standing	1	0	0	15	516	0	97
Laying	0	0	0	0	0	540	100
Precision (%)	98	97	96	97	94	99	

(a)



(b)

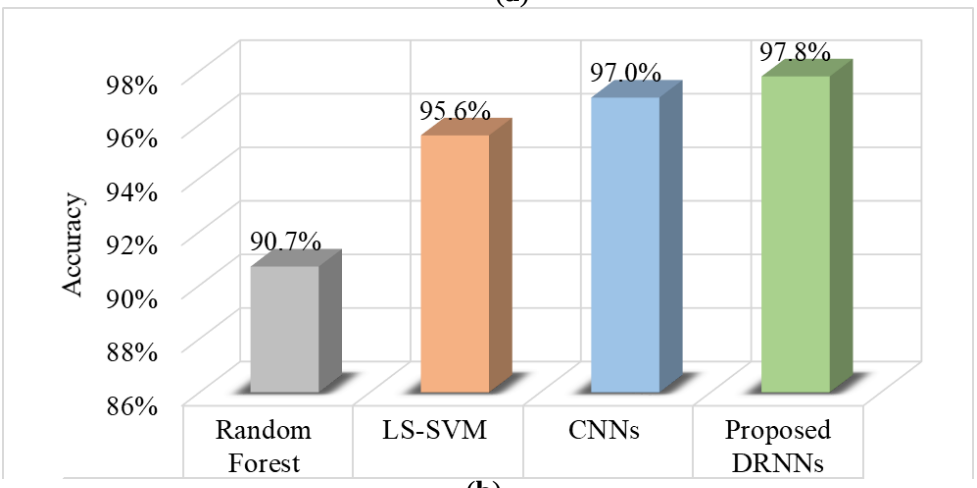
Figure 14. Performance results of the proposed unidirectional DRNN model for the UCI-HAD dataset: (**a**) Confusion matrix for the test set containing the activity recognition results. The rows represent the true labels and the columns represent the model classification results; (**b**) Comparative accuracy of the proposed model against other methods.

b) USC-HAD

We found that the unidirectional DRNN model with four layers yields the best results for the USC-HAD dataset. Figure 15a presents the classification results for the test set in the form of a confusion matrix, along with the per-class recall and precision results. The proposed method achieved better overall accuracy than other methods, such as CNN [53], least squares support vector machines (LS-SVM) [55], and random forest [56], as shown in Figure 15b.

	W. Forward	W. Left	W. Right	W. Upstairs	W. Downstairs	Running	Jumping	Sitting	Standing	Sleeping	In Elevator	**Recall (%)**
W. Forward	1576	8	7	3	5	0	0	0	8	0	0	98
W. Left	11	1060	4	0	6	1	0	2	1	0	0	98
W. Right	8	4	1095	2	1	0	0	0	4	0	0	98
W. Upstairs	1	1	7	889	3	0	2	0	4	0	0	98
W. Downstairs	1	4	3	1	840	0	12	0	1	0	0	97
Running	3	1	2	2	2	713	1	0	2	0	0	98
Jumping	1	1	1	2	5	0	412	1	4	0	1	96
Sitting	0	0	0	0	0	0	1	1008	13	0	2	98
Standing	1	0	0	1	1	0	1	11	940	0	34	95
Sleeping	0	0	0	0	0	0	0	0	1	1587	0	100
In Elevator	0	0	0	0	0	0	0	1	37	0	632	94
Precision (%)	98	98	98	99	97	100	96	99	93	100	94	

(a)



(b)

Figure 15. Performance results of the proposed unidirectional DRNN model for USC-HAD dataset: (**a**) Confusion matrix for the test set displaying activity recognition results with per-class precision and recall; (**b**) Comparative accuracy of proposed model against other methods.

c) Opportunity

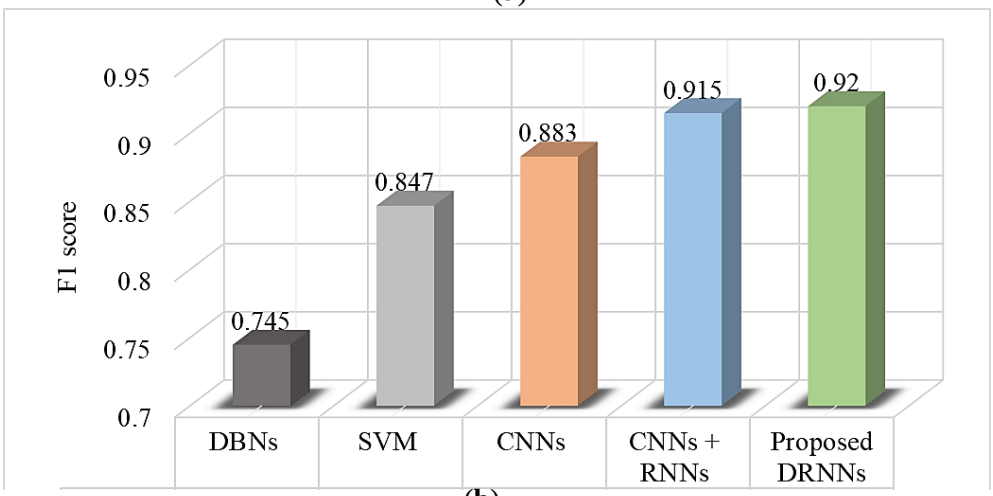
The Opportunity dataset is very complex and contains a wide range of activities. Therefore, the bidirectional DRNN model with three layers yields the best performance results. The confusion matrix in Figure 16a summarizes the classification results of the proposed model for the test set, along with the per-class precision and recall results. The proposed method outperforms other methods, such as those based on deep believe networks (DBN) [33], SVM [33], and CNN [40]. It also outperformed the state-of-the-art method, which is a combination of CNN and unidirectional RNN [40], for the opportunity dataset. Figure 16b presents a performance comparison between the F1 score of the proposed method and those reported by other methods. We used the F1 score as a basis for comparison because the Opportunity dataset is imbalanced, manifested by the dominance of the Null class.

d) Daphnet FOG

For the Daphnet FOG dataset, we found that the cascaded DRNN model with one bidirectional layer and two upper unidirectional layers yields the best results. Figure 17a summarizes the classification results for the test set. The low values of recall and precision for the “Freeze” class are caused by the dominance of the “Normal” class. However, our proposed method still outperforms other methods, such as k-nearest neighbors (KNN) [57] and CNN [58], in terms of F1 score, as shown in Figure 17b.

	Null	Open Door1	Open Door2	Close Door1	Close Door2	Open Fridge	Close Fridge	Open Dishwasher	Close Dishwasher	Open Drawer 1	Close Drawer 1	Open Drawer 2	Close Drawer 2	Open Drawer 3	Close Drawer 3	Clean Table	Drink from Cup	Toggle Switch	**Recall (%)**
Null	9347	12	8	14	7	25	30	10	38	8	16	12	7	11	16	16	90	5	97
Open Door 1	9	178	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92
Open Door 2	13	1	229	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	93
Close Door 1	6	11	0	201	1	0	0	0	0	0	0	0	0	0	0	0	0	0	92
Close Door 2	10	0	7	2	216	0	0	0	0	0	0	0	0	0	0	0	0	0	92
Open Fridge	97	0	0	0	0	224	12	3	0	4	0	0	2	0	0	0	0	0	65
Close Fridge	27	0	1	0	0	6	234	1	1	0	1	0	0	0	0	0	1	0	86
Open Dishwasher	53	0	0	0	0	4	3	175	0	0	0	0	0	0	0	0	0	0	74
Close Dishwasher	27	0	0	0	0	0	2	2	167	0	0	1	3	2	3	0	0	0	81
Open Drawer 1	15	0	0	0	0	0	0	2	0	105	4	4	2	0	2	0	0	2	77
Close Drawer 1	19	0	0	0	0	0	2	0	0	5	101	2	2	0	0	0	0	0	77
Open Drawer 2	12	0	0	0	0	1	0	0	0	4	0	116	3	2	0	0	0	2	83
Close Drawer 2	9	0	0	0	0	0	0	1	0	0	0	5	108	0	0	0	0	0	88
Open Drawer 3	12	0	0	0	0	0	0	2	0	1	0	7	3	161	3	0	0	0	85
Close Drawer 3	9	0	0	0	0	0	0	0	2	0	0	2	6	9	146	0	0	0	84
Clean Table	44	0	0	0	0	4	0	0	0	0	0	1	0	2	0	229	0	0	82
Drink from Cup	105	1	0	0	0	1	0	3	0	0	0	0	0	0	0	0	772	0	88
Toggle Switch	56	1	0	0	0	0	0	0	0	6	4	0	0	0	0	0	0	150	69
Precision (%)	95	87	93	90	95	85	83	88	80	79	80	77	79	86	86	93	89	94	

(**a**)

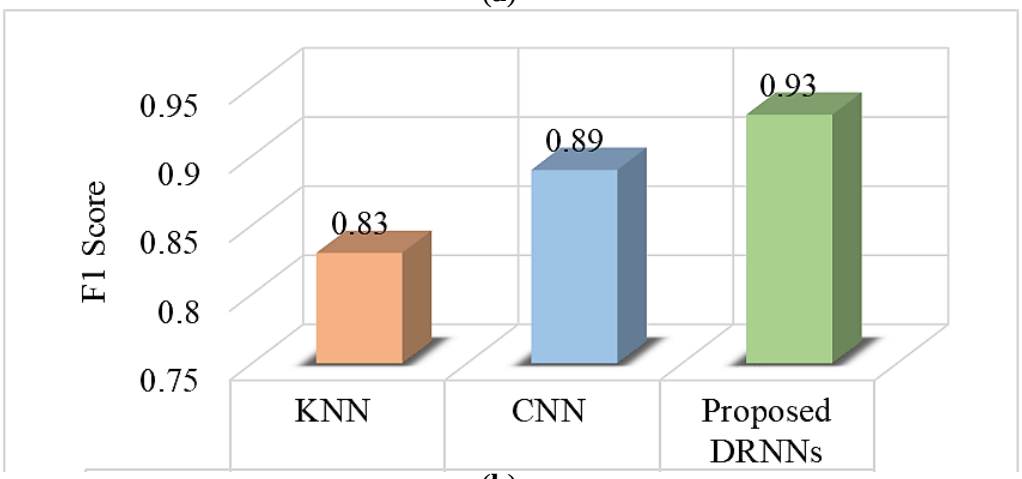


(**b**)

Figure 16. Performance results of the proposed bidirectional DRNN model for the Opportunity dataset: (**a**) Confusion matrix for the test set as well as per-class precision and recall results; (**b**) Comparative F1 score of proposed model against other methods.

	Freeze	Normal	**Recall (%)**
Freeze	822	545	60.1
Normal	295	12591	97.7
Precision (%)	73.6	95.9	

(a)



(b)

Figure 17. Performance results of the proposed cascaded DRNN model for the Daphnet FOG dataset: (a) Confusion matrix for the test set, along with per-class precision and recall; (b) F1 score of the proposed method in comparison with other methods.

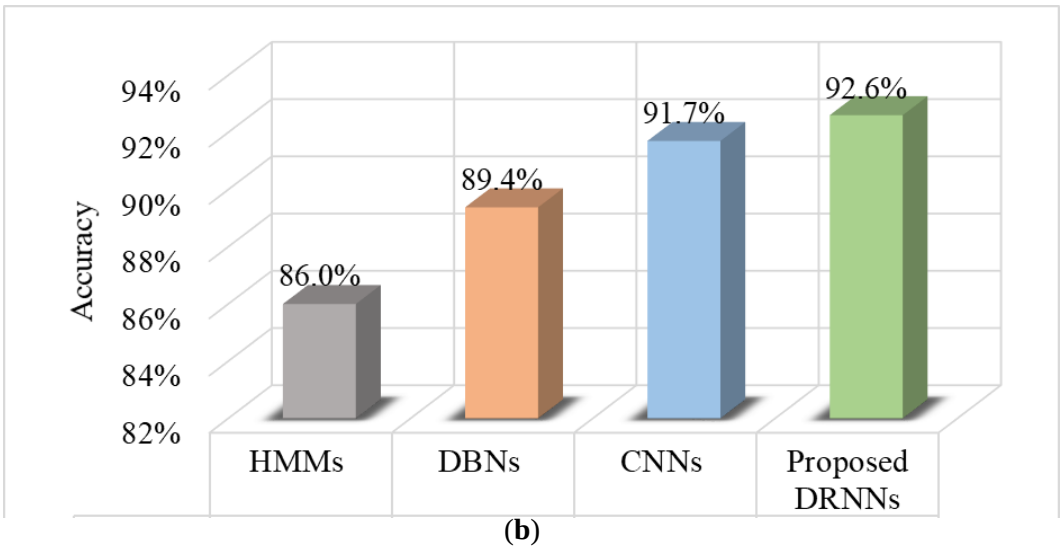
e) Skoda

We found that the cascaded DRNN model yields the best results for the Skoda dataset. The model is built using one bidirectional layer and two upper unidirectional layers. Figure 18a presents the classification results for the test set in the form of a confusion matrix, along with the per-class recall and precision results. The proposed method results in an overall accuracy of

92.6%, outperforming other methods such as HMMs [49], DBNs [27], and CNN [34], as shown in Figure 18b.

	Null	Write on notepad	Open hood	Close hood	Check gaps on front door	Open left front door	Close left front door	Close both left door	Check trunk gaps	Open and close trunk	Check steering wheel	**Recall (%)**
Null	234	1	6	6	0	1	2	4	0	2	2	91
Write on notepad	4	106	1	0	0	0	0	1	0	0	0	95
Open hood	3	0	93	4	2	0	0	1	0	1	1	88
Close hood	2	0	2	90	1	0	0	1	0	2	0	92
Check gaps on front door	2	0	0	0	77	0	0	0	1	0	0	96
Open left front door	5	0	0	0	0	34	1	0	0	1	0	83
Close left front door	3	0	0	0	0	1	43	0	0	0	0	91
Close both left door	0	0	0	2	0	0	0	82	1	0	0	96
Check trunk gaps	0	0	0	0	2	0	0	3	77	0	0	94
Open and close trunk	2	0	1	0	0	0	1	0	0	96	0	96
Check steering wheel	2	0	0	0	0	0	0	0	0	0	47	96
Precision (%)	91	99	90	88	94	94	91	89	97	95	94	

(a)



(b)

Figure 18. Performance results of the proposed cascaded DRNN model for the Skoda dataset: (a) Confusion matrix for the test set as well as per-class precision and recall results; (b) Comparative accuracy of proposed model against other methods.

7.3 Discussion

The performance results of the proposed models clearly demonstrate that DRNN are very effective for HAR. All of the architectures performed very well on all of the datasets. These datasets are diverse, which proves that our models are effective for a broad range of activity recognition tasks. The unidirectional DRNN model yielded the best results for the UCI-HAD and USC-HAD datasets, the bidirectional DRNN model gave better results for the Opportunity dataset, and the cascaded DRNN model performed better on the Daphnet FOG and Skoda dataset. Table 2 contains a performance summary for the four datasets.

There are two main reasons for the superb performance of the proposed models for HAR tasks [59]. First, including sufficient deep layers enabled the models to extract effective discriminative features. These features are exploited to distinguish between classified activities and scale up for more complex behavior recognitions tasks. Second, employing DRNN to capture sequential and time dependencies between input data samples provided a significant improvement in performance compared to other methods.

Table 2. Performance summary for the proposed DRNN on five diverse datasets.

Model	Dataset	Overall Accuracy	Average Precision	Average Recall	F1 score
Unidirectional DRNN	UCI-HAD	96.7%.	96.8%,	96.7%	0.96
Unidirectional DRNN	USC-HAD	97.8%	97.4.0%	97.4%	0.97
Bidirectional DRNN	Opportunity	92.5%	86.7%	83.5%	0.92
Cascaded DRNN	Daphnet FOG	94.1%	84.7%	78.9%	0.93
Cascaded DRNN	Skoda	92.6%	93.0%	92.6%	0.92

Chapter 7: Conclusion

In this thesis, we have presented a novel HAR system that performs direct end-to-end mapping from raw sensors data to activity label classifications. The system is built using LSTM-based DRNN with three viable architectures: unidirectional, bidirectional, and cascaded DRNN models. The proposed models are based only on DRNN, meaning we avoid the complexity of combining multiple deep learning approaches in a single framework. Additionally, by using only DRNN, our models are more flexible for classifying variable-length windows, in contrast to the fixed-length windows required by CNN.

We empirically evaluated our models by conducting experiments on five miscellaneous benchmark datasets. Experimental results revealed that the proposed models outperform other state-of-the-art methods. The reason for this improvement in performance is that our models are able to extract more discriminative features by using deep layers in a task-dependent and end-to-end fashion. Furthermore, our models are able to capture the temporal dependencies between input samples in activity sequences by exploiting DRNN functionality.

Future research in HAR based on the work presented in this thesis includes experimentation on large-scale and complex human activities, as well as exploring transfer learning between diverse datasets. Investigating resource efficient implementation of a DRNN for low-power devices is also a promising future research direction. In addition, further work would focus more on extracting qualitative information assessing the quality of an activity.

Bibliography

[1] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.

[2] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM Neural Networks for Language Modeling," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[3] L. Yao, K. Cho, N. Ballas, C. Paí, and A. Courville, "Describing Videos by Exploiting Temporal Structure," in *IProceedings of the IEEE international conference on computer vision*, 2015.

[4] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, vol. 385. Berlin, Heidelberg: Springer, 2012.

[5] T. Mikolov, M. Karafiát, L. Burget, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[6] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies," in *Field Guide to Dynamical Recurrent Networks*, 1st ed., S. Kremer and J. Kolen, Eds. Wiley-IEEE Press, 2001, p. 237–243,.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[8] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.

[9] F. Foerster, M. Smeja, and J. Fahrenberg, “Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring,” *Comput. Human Behav.*, vol. 15, no. 5, pp. 571–583, Sep. 1999.

[10] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork, “Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity,” in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008, pp. 5250–5253.

[11] M. Ermes, J. Parkka, and L. Cluitmans, “Advancing from offline to online activity recognition with wearable sensors,” in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008, pp. 4451–4454.

[12] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, “Real-Time Recognition of Physical Activities and Their Intensities Using Wireless Accelerometers and a Heart Rate Monitor,” in *2007 11th IEEE International Symposium on Wearable Computers*, 2007, pp. 1–4.

[13] L. Bao and S. S. Intille, “Activity Recognition from User-Annotated Acceleration Data,” in *2008 International Conference on Machine Learning and Cybernetics*, 2004, pp. 1–17.

[14] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, “Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions,” in *International Workshop on Wearable and Implantable Body Sensor Networks (BSN’06)*, pp. 113–116.

[15] Zhen-Yu He and Lian-Wen Jin, “Activity recognition from acceleration data using AR model representation and SVM,” in *2008 International*

Conference on Machine Learning and Cybernetics, 2008, pp. 2245–2250.

[16] Zhenyu He, Zhibin Liu, Lianwen Jin, Li-Xin Zhen, and Jian-Cheng Huang, "Weightlessness feature — a novel feature for single tri-axial accelerometer based activity recognition," in *2008 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[17] Z. He and L. Jin, "Activity recognition from acceleration data based on discrete consine transform and SVM," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 5041–5044.

[18] T.-P. Kao, C.-W. Lin, and J.-S. Wang, "Development of a portable activity detector for daily activity recognition," in *2009 IEEE International Symposium on Industrial Electronics*, 2009, pp. 115–120.

[19] Y.-P. Chen, J.-Y. Yang, S.-N. Liou, G.-Y. Lee, and J.-S. Wang, "Online classifier construction algorithm for human activity detection using a tri-axial accelerometer," *Appl. Math. Comput.*, vol. 205, no. 2, pp. 849–860, Nov. 2008.

[20] A. Zinnen, C. Wojek, and B. Schiele, "Multi Activity Recognition Based on Bodymodel-Derived Primitives," in *International Symposium on Location- and Context-Awareness*, 2009, pp. 1–18.

[21] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies - sOc-EUSAI '05*, 2005, p. 159.

[22] A. Bulling, C. Weichel, and H. Gellersen, "EyeContext," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, 2013, p. 305.

[23] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, "Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2521–2537, Dec. 2011.

[24] P. Vepakomma, D. De, S. K. Das, and S. Bhansali, "A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities," in *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2015, pp. 1–6.

[25] K. H. Walse, R. V. Dharaskar, and V. M. Thakare, "PCA Based Optimal ANN Classifiers for Human Activity Recognition Using Mobile Sensors Data," Springer, Cham, 2016, pp. 429–436.

[26] T. Plötz, N. Y. Hammerla, and P. Olivier, "Feature Learning for Activity Recognition in Ubiquitous Computing," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, 2011, vol. 2, pp. 1729–1734.

[27] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep Activity Recognition Models with Triaxial Accelerometers," in *AAAI Workshop: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, 2016.

[28] B. Almaslukh, J. Almuhtadi, and A. Artoli, "An Effective Deep Autoencoder Approach for Online Smartphone- Based Human Activity Recognition," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 17, no. 4, 2017.

[29] A. Wang, G. Chen, C. Shang, M. Zhang, and L. Liu, "Human Activity Recognition in a Smart Home Environment with Stacked Denoising Autoencoders," Springer, Cham, 2016, pp. 29–40.

[30] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors," in *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*, 2014, pp. 197–205.

[31] Y. Chen and Y. Xue, "A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 1488–1492.

[32] H.-O. Hessen and A. J. Tessem, "Human Activity Recognition with two Body-Worn Accelerometer Sensors," MS thesis, Norwegian University of Science and Technology, Norway, 2015.

[33] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina*, 2015, no. Ijcai.

[34] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2016, pp. 71–76.

[35] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust.*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

[36] C. Liu, L. Zhang, Z. Liu, K. Liu, X. Li, and Y. Liu, "Lasagna: towards deep hierarchical understanding and searching over mobile sensing data," in *Proceedings of the 22nd Annual International Conference on*

Mobile Computing and Networking - MobiCom '16, 2016, pp. 334–347.

[37] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification," *Front. Comput. Sci.*, vol. 10, no. 1, pp. 96–112, Feb. 2016.

[38] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential Deep Learning for Human Action Recognition," in *International Workshop on Human Behavior Understanding*, 2011, pp. 29–39.

[39] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, K. S. Umass, L. Lowell, and T. Darrell, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," in *IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[40] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[41] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based Recurrent Neural Networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014, no. September, pp. 1964–1968.

[42] J. Kittler, M. Hater, and R. P. W. Duin, "Combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, no. 3, pp. 226–239, 1996.

[43] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[44] Y. Wu and . et. al, "Google's Neural Machine Translation System:

Bridging the Gap between Human and Machine Translation," *CoRR*, 2016.

[45] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," in *European Symposium on Artificial Neural Networks*, 2013, no. April, pp. 24–26.

[46] M. Zhang and A. A. Sawchuk, "USC-HAD: A Daily Activity Dataset for Ubiquitous Activity Recognition UsingWearable Sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1036–1043.

[47] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. D. R. Millán, and D. Roggen, "The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, 2013.

[48] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Troster, "Wearable assistant for Parkinson's disease patients with the freezing of gait symptom," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 436–446, 2010.

[49] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection," in *Wireless Sensor Networks*, Berlin, Germeny: Springer Berlin Heidelberg, 2008, pp. 17–33.

[50] I. Goodfellow, Y. Bengio, and A. Courville, "Optimization for Training Deep Models," in *Deep Learning*, Cambridge, Massachusetts, USA: The MIT Press, 2016, p. 800.

[51] M. Abadi and . et. al, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.” Software available from tensorflow.org, p. Software available from tensorflow.org, 14-Mar-2015.

[52] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, “Dropout Improves Recurrent Neural Networks for Handwriting Recognition,” in *14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 285–290.

[53] W. Jiang, “Human Activity Recognition using Wearable Sensors by Deep Convolutional Neural Networks,” in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1307–1310.

[54] R. Chandan Kumar, S. S. Bharadwaj, B. N. Sumukha, and K. George, “Human activity recognition in cognitive environments using sequential ELM,” in *Second International Conference on Cognitive Computing and Information Processing*, 2016, pp. 1–6.

[55] Y. Zheng and Yuhuang, “Human Activity Recognition Based on the Hierarchical Feature Selection and Classification Framework,” *J. Electr. Comput. Eng.*, vol. 2015, p. 34, 2015.

[56] B. B. Prakash Reddy Vaka, “A Pervasive Middleware for Activity Recognition with smartphones,” MS Thesis, University of Missouri, US, 2015.

[57] N. Hammerla and R. Kirkham, “On Preserving Statistical Characteristics of Accelerometry Data using their Empirical Cumulative Distribution,” in *Proceedings of the 2013 International Symposium on Wearable Computers*, 2013, pp. 65–68.

[58] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, “A deep learning approach to on-node sensor data analytics for mobile or wearable devices,” *IEEE J. Biomed. Heal. Informatics*, vol. 21, no. 1, pp. 56–64, 2017.

[59] A. Murad and J.-Y. Pyun, “Deep Recurrent Neural Networks for Human Activity Recognition,” *Sensors*, vol. 17, no. 11, p. 2556, Nov. 2017.