IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 3, March 2020

Recognizing Activities of Daily Living in Ambient Intelligence Environments using 1D Convolutional Neural Networks

Sumaya Alghamdi[1], Etimad Fadel[2], Nahid Alowidi[3]

Student, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, KSA[1]

Associate Professor, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, KSA[2]

Assistant Professor, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, KSA[3]

Abstract: Human Activity Recognition (HAR) is considered a challenging task in sensor-based monitoring systems. In ambient intelligent environments, such as smart homes, collecting data from multiple sensors is useful for recognizing Activities of Daily Living (ADLs), which can then be used to help provide assistance to inhabitants. ADLs are composed of complex time-series data that has high dimensionality, is large in size, and is updated continuously. Thus, developing methods for analysing these time-series data to extract meaningful features and specific characteristic would help solve the problem of activity recognition. Based on the noticeable success of deep learning in the time-series classification field, we developed a model for classifying ADLs in an ambient environment using deep neural networks. Our model, a Deep One-Dimensional Convolutional Neural Network (Deep 1D-CNN), contains several one-dimensional convolution layers coupled with a max-pooling technique to discover and extract the suitable internal structure to generate the deep features of the input time-series automatically. Such a model can be used as a unified framework for both feature extraction and classification. It performs well on high-dimensional time-series data; it does not require any expert knowledge in feature extraction, and it is able to find relevant and discriminative features for activity recognition. In order to evaluate the performance of our model, we tested it on the new real-life dataset, ContextAct@A4H, and the results showed that our model achieved a high F1 score (0.90). We also compared our results with baseline models for time series classification with deep neural networks. The comparison revealed that, our deep 1D-CNN model achieved the best overall performance in terms of precision, recall, and F1 score.

Keywords: Deep Learning, One-Dimensional Convolutional Neural Networks, Time-series Classification, Activities of Daily Living (ADLs), smart home

I. INTRODUCTION

In recent decades, we have seen a noticeable increase in the number of Internet of Things (IoT) applications, such as are found in smart homes. These environments rely on using sensors to generate large amounts of time-series data that can be analyzed for many purposes, such as the monitoring, detection, and classification of activities in order to make timely decisions [1]. Recognizing human activity in an automated manner has become essential in many ambient intelligence applications, and a smart home with Artificial Intelligence (AI) is a great step towards it [2].

Human Activity Recognition (HAR) at smart home can be categorised into sensor-based activity recognition and vision based activity recognition. Sensor based activity recognition can be done using two kind of sensors which are wearable sensors on a human body to recognize frequent motions like walking and standing, and ambient sensors for recognizing Activities of Daily Living (ADLs) to predict the complex activities of humans, such as watching TV and cooking. The main difference between these kinds is that sensor based ambient systems have the ability to recognize different behaviours and complex activities over time by monitoring the interaction between objects and people in smart environments while body worn sensors can only capture human physical activities [3]. Using ambient sensors for recognizing ADLs in smart home can provide many benefits for people who live there. One of these benefits is to predict which activity will be performed by inhabitant, thus, the system can automatically provide needed services [4]. Another one is increasing the potential of energy saving by using the sensors to monitor the energy consumption of devices at smart environment and reduce it either by controlling them directly or sending recommendations to users on how it is possible to save energy at a particular time [5]. Therefore, human activity recognition and a learning process in a smart home is necessary and important.

Recently, increased attention has been given to deep learning in many fields, such as speech recognition [6] and image classification [7]. In activity recognition, using deep learning can have a great impact in terms of system performance and flexibility. Such methods can provide efficient tools for high-level feature extraction from high-dimensional time-series data, which is useful for many tasks such as classification. Deep-learning models have demonstrated the significant

DOI 10.17148/IJARCCE.2020.9301

potential to push the state of the art in activity recognition. They overcome the need for feature engineering that refers to defining feature extraction procedures manually, which is usually a time-consuming task and can be error-prone or (at least) a poorly generalizable endeavor. Generally, deep learning refers to neural networks that utilize many layers for processing non-linear information organized hierarchically, where the output for each layer is the input for the next one. The well-known deep-learning architectures include Convolutional Neural Networks (CNNs) [8], Recurrent Neural Networks (RNNs) [9], and Deep Belief Networks (DBNs) [10].

Since the multi-sensor data of ADLs is high-dimensional data with significant variation, applying feature selection methods on the data before feeding it into the deep-learning networks would solve the complexity of activity recognition and increase the accuracy for such a system. Indeed, feature selection is a fundamental step when dealing with high-dimensional data, allowing for the elimination of those variables that are redundant or irrelevant for the system description [11]. In addition, deep-learning models (i.e., CNNs) have been shown capable and even achieve better state-of-the-art results for time-series classification by automatically capturing local dependencies and preserving feature scale invariance to catch variation in the same class throughout the feature extraction [12]. Thus, in this paper, we take advantage of feature importance as a feature selection technique combined with deep One-Dimensional (1D) CNNs, since CNNs in general are good at reducing frequency variations and capturing local dependency. We explored our model, which we call deep 1D-CNN, on the new large dataset ContextAct@A4H [13] that collected from an ambient environment to determine the capacity for such a model when dealing with long input sequences to recognize complex activities, such as sleeping and watching TV.

The rest of this paper is organized as follows. In Section II, we provide a primer on the relevant background in CNNs. The existing works on the deep-learning-based recognition of ADLs in ambient environments are reviewed in Section III. A detailed description is presented in Section IV, illustrating the structure of our proposed architecture. The model implementation details are demonstrated in Section V with the experimental setup. Then, the results and the performance comparisons are presented in Section VI. Finally, the conclusion comes in Section VII.

II. BACKGROUND

Neural networks are suitable for time-series classification tasks, and they are the basis of deep-learning techniques. As mentioned previously, recognizing ADLs using deep learning has become one of the most preferred techniques, owing to its ability to effectively learn data representation and perform classification. One of the most recent popular deep-learning methods for time-series classification is One-Dimensional Convolutional Neural Network (1D-CNN), which is the most powerful deep network for feature learning. In what follows, we briefly summarize the architecture of Convolutional Neural Networks (CNNs) that are the bias of our model.

A. Convolutional Neural Networks (CNNs)

CNNs are a type of deep neural network, composed of multiple hidden layers of convolution, activation, and pooling layers. A single layer of the CNN extracts features through convolutional operations that are applied on input sequences. The output of the convolutional operation is then passed through the non-linear activation function and then fed to the pooling layers [9]. Formally, extracting a feature map using a convolution operation is given by:

$$f^j = \sigma(b^i + \sum_i k^{ij} * x^i) \quad (1)$$

where x^i *and* f^j is the *i*-th and *j*-th input and output maps, and k^{ij} is the convolution kernel that slides from the start-time step to the end for completing the convolutional operations on the entire input sequence. The * is the convolution operator, and σ is the activation function, “which is in most cases ReLU,” and it is computed by the following equation:

$$ReLU(x) = max(\{0.x\}) \quad (2)$$

The non-linearity activation function is usually followed by pooling layers. One of the most used pooling technique is a max-pooling layer, which performs a maximum subsampling of the feature maps obtained in the previous layer. Max-pooling layers are used to reduce the spatial size of the representation to minimize the amount of parameters and computation in the network, and hence, also control overfitting. The max-pooling can be identified as:

$$p^i = max\ (a.c)\ where\ a.c \in f^j \quad (3)$$

where the vector p^i is the *i*-th output of the max-pooling layer and f^j is the input feature map obtained from Equation (1). The combination of convolutional, activation, and pooling layers allows the CNNs to learn and extract deep features that lead to an extremely effective system. CNN models with several layers, where the input of layer (*i* – 1) is the input of layer *i*, have the ability to learn data representations for faster feature extraction and classification. Such a model with deeper layers has an advantage over other networks in the field of time-series classification for activity recognition [12], where it has become the standard approach.

III. RELATED WORK

In the field of human activity recognition, the use of sensors is extremely diverse. As a matter of fact, activity recognition is used in wearable devices to recognize human body movements such as walking and running [14] or in smart industrial fields for complex activity recognition using ambient sensors [15]. Among the various activity recognition fields, we are focused on reviewing recent studies on daily-activity recognition-based ambient sensors, specifically, smart homes by applying deep neural-network-based approaches.

In general, the input of the neural network is the original raw data; applying features extracted from these data to the neural network tend to improve the overall performance. Extracting useful information from the original data requires expert knowledge, however, convolutional neural networks have been proposed to address this problem. CNNs are one of the most widely used deep-learning approaches in activity recognition fields not only for extracting features but also for activity classification. In ambient-based environments, CNN with 1D kernal have been used to extract features from data obtained from multiple ambient sensors for daily activity classification. In [16], the authors proposed architecture-based CNNs for solving the complexity of recognizing ALDs. With a one-dimensional convolution layer for feature extraction equipped with a fully connected layer and softmax classifier, their model achieved an accuracy of about 0.72 on the Kasteren dataset. Another work [17] evaluated several state-of-the-art machine-learning approaches and deep learning models including CNN for classifying daily activities based on ambient sensors. Overall, model based CNNs showed the best performance in terms of the F1 score on the real-world dataset ARAS when compared to other classification approaches.

Since the data of daily activities are sequences recorded by multiple sensors, recurrent-based neural networks have been widely proposed to solve the complexity of sequence-lapping problems in daily activity recognition. Paper [18] investigated three variants of recurrent neural networks (RNNs), which were vanilla RNNs (VRNN), long short-term memory RNNs (LSTM) and gated recurrent unit RNNs (GRU). They compared them with the state-of-the-art methods, such as supported vector machines (SVMs) and hidden Markov models (HMMs). They used the dataset collected by Kasteren to evaluate all three models. The results indicated that the models based on RNNs were competitive with the state-of-art methods; however, LSTM seemed to be slightly better. Another paper [19] applied LSTM-RNNs on three real-world smart-home datasets for recognizing daily activities. With one LSTM layer equipped with 300 hidden units, their model achieved an accuracy of about 79.90. The results showed that the LSTMs outperformed other state-of-the-art methods such as naive Bayes and HMMs. Recently, paper [20] analyzed the sequence of actions recorded in a smart home with ambient multiple sensors by introducing the concept of residual recurrent neural network (Residual-RNN). Their experiments evidence that the proposed model performed better than the base LSTMs and gated recurrent units (GRU) in respect to recognition accuracy.

Some works solved the complexity of recognizing ADLs with sensor data collected in ambient-based environments by applying simple neural network algorithms, such are deep meta-layer networks [21] and Levenberg-Marguardt neural networks [22]. However, such simple networks can perform well only on small datasets due to their simple architectures.

To sum up, it is noticeable that the common efficient deep-learning approaches used for recognizing ADLs in an ambient intelligent environment are both 1D-CNNs and recurrent-based networks such as LSTMs. Since we are dealing with the classifying time-series of sequence data collected from sensors in an ambient environment, CNNs are capable of discovering and extracting the suitable internal structure to generate deep features of the input sequence automatically to recognize ADLs. In the next section, the architecture of our proposed model (deep 1D-CNN) is described in detail.

IV. METHODOLOGY

A. Overview

We propose a deep-learning model for time-series classification to recognize activities of daily living. The deep architecture of the proposed model contains multiple one-dimensional convolution and max-pooling layers.

Before modelling our data, a feature selection method is applied for identifying the related features from a set of data and removing the less important features (see Section V-B for more detail). The input of the networks is sequences of time-series data obtained from the multi-sensors using a segmentation-based approach. The time-series segmentation method we applied is a fixed-sized overlapping sliding window that divides the data into multiple sequences of discrete segments that share the same length and are assigned to different labels (see Section V-C for more detail). Those fixed-length sequences will feed the input of our model to train it for the classifying daily activities, such as cooking, eating, and working.

In what follows, we will describe in detail the proposed architecture.

B. Deep 1D-CNN

Current common algorithms mainly use CNNs for feature extraction and classification, yet, these methods are often not popular in dealing with time-series data. We propose a deep-learning method for activity recognition that is able to handle input sequences of time-series data obtained from ambient intelligent environments.

DOI 10.17148/IJARCCE.2020.9301

Suppose the i^{th} input time series is:

$$\mathrm{C_i} = [\, \mathrm{c_i^{(1)}} \,.\, \mathrm{c_i^{(2)}} . \ldots\ldots . \mathrm{c_i^{(t)}} . \ldots\ldots . \mathrm{c_i^{l}} \,].\ \mathrm{C_i} \in \mathbb{R}^{\,\mathrm{d}} \qquad (4)$$

where l is the length of the sequence, $c_i^{(t)}$ denotes the value of time step *t*, and *d* represents the input dimension (i.e., the number of input variables).

The main component of the proposed model is the one-dimensional CNNs. Their purpose is to automatically extract deep and short-term features from the multivariable time-series input. We built our model by stacking multiple convolutional, batch normalization, max-pooling, and dropout layers. At first, the convolution operations are applied on the sequence input to obtain the feature mapping, which was formalized in Equation (1). Before the activation function and right after the convolutional layer, Batch Normalization (BN) [23] is implemented to accelerate the training process and improve the efficiency. Then, the rectified linear unit (ReLU) activation function is employed to insure the non-linear behavior of the network. Max-pooling layers are used to subsample the extracted feature mapping, which helps eliminate non-maximal values and extract the local dependency within different regions to keep the most salient information. After the max-pooling layers, dropout is applied to avoid overfitting and decrease training time. After the layers have extracted relevant features from all the input sequence data, the results will be flattened into a vector. The final obtained feature mapping is then fed to a softmax layer (i.e., a dense layer with a softmax activation function) to obtain the output by yielding a class probability distribution for every single time step t. By stacking several convolutional operators with the max-pooling technique, the proposed model has the ability to extract the deep features from the input sequence to create a hierarchy of progressively abstract features that are more robust to noise in order to classify each activity successfully. Both a batch normalization and dropout are applied to each 1D-CNN layer to improve the overall performance. The proposed approach is defined according to the network structure depicted in Fig. 1.

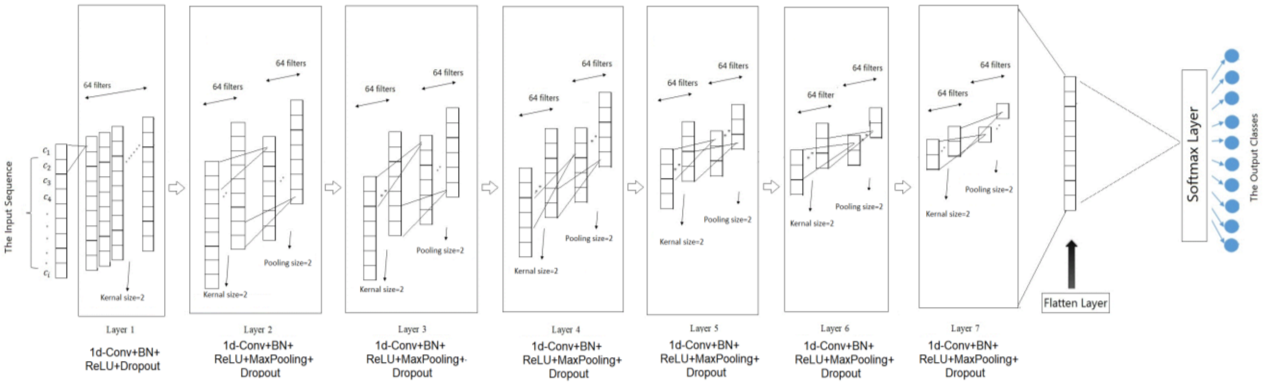
The Input Sequence
c_1
c_2
c_3
c_4
c_l
64 filters
Kernal size=2
Pooling size=2
64 filter
Layer 1
1d-Conv+BN+ ReLU+Dropout
Layer 2
1d-Conv+BN+ ReLU+MaxPooling+ Dropout
Layer 3
1d-Conv+BN+ ReLU+MaxPooling+ Dropout
Layer 4
1d-Conv+BN+ ReLU+MaxPooling+ Dropout
Layer 5
1d-Conv+BN+ ReLU+MaxPooling+ Dropout
Layer 6
1d-Conv+BN+ ReLU+MaxPooling+ Dropout
Layer 7
1d-Conv+BN+ ReLU+MaxPooling+ Dropout
Flatten Layer
Softmax Layer
The Output Classes

Fig. 1 The architecture of deep 1D-CNN framework for daily activity recognition.

V. IMPLEMENTATION

This section extensively discusses the dataset and the pre-processing steps we applied to prepare it for further analysis. It also provides an overview of the experimental environment, followed by the parameters set for our model.

A. Dataset

We evaluated our deep 1D-CNN model on the new real-world dataset ContextAct@A4H. This is a publicly available rich dataset collected from multiple ambient sensors that reflect the context of the environment as well as annotated standard daily living activities. It is a real-life dataset that was gathered at the Amiqual4Home apartment, using a large variety of sensors to improve the potential using the data. Ambient sensors were used to allow observation of both object usage and context conditions for all rooms and the exterior. The following context variables were measured in each room: temperature, carbon dioxide (CO2), noise, humidity, presence, and music information. Weather information was measured for the exterior. Appliance and object usage were measured through electric/water consumption sensors, contact sensors, and state change sensors. Other sensors were indirect measures of object usage (e.g., pressure sensor in the couch).

The dataset consisted of one week of captured data during the summer and three weeks in the fall. It contains ten daily activities were annotated by self-reporting in place at the moment of starting and ending each activity using three different annotating methods. The activities and their annotated names with the assigned labels and number of examples per each are listed in Table I

DOI 10.17148/IJARCCE.2020.9301

Table I Annotated Activities in Contextact@A4h

Activity	Annotated as…	Labelled as..	# of Examples
Working	Travail	0	38,369
Cooking	Cuisine	1	50,969
Relaxing	Loisir	2	4536
Using the Toilet	Toilettes	3	8569
Watching TV	TV	4	22,085
Washing Dishes	Vaisselle	5	16,045
Leaving Home	Sortir	6	213,665
Taking a Shower	Douche	7	35,993
Eating	Manger	8	41,195
Sleeping	Dormir	9	210,566

This dataset contains about 389 features, some of them are not used in the case of recognizing ADLs (e.g., music features). Thus, in the pre-processing phase, we assigned all the values for those unneeded features to zero. In addition, we converted all needed contextual features to numerical information. For example, binary sensors such as contact sensors give two values, such as 'open' or 'closed' to determine if a door has been opened or has not. In those cases, we converted the measured unit from contextual information to be either 1 or 0. Moreover. We had to clean corrupted and repeated data that could have been caused by possible hardware failures or problems in data transfer. As with any data pre-processing, we applied data normalization, which is the process of rescaling all the feature values to be in a given range on the dataset. In particular, we used a min-max scaler, which changed all the features to be between 0 and 1. After the data normalization, we split the data randomly into 70% for training and 30% as the test set. After making the dataset suitable for further analysis, we applied a feature selection method that is explained in the next subsection.

B. Feature Selection

In many practical applications of machine-learning algorithms, the data that has been pre-processed can result in a large number of features, and it is often desirable to reduce this number to enable the effective classification beyond the training stage. Furthermore, supervised learning models that have been trained on data incorporating many features may suffer from overfitting the model to the training data. As we mentioned previously, the dataset had about 389 features of sensor readings that ranged between 0 to 1. Our purpose was to reduce the number of features and choose only those which contributed the most in increasing the overall performance of our classification models. For such an aim, we applied a common feature selection technique that extracted a feature importance rank using different classifiers such as adaptive boosting (AdaBoost) [24] as shown in Fig. 2.

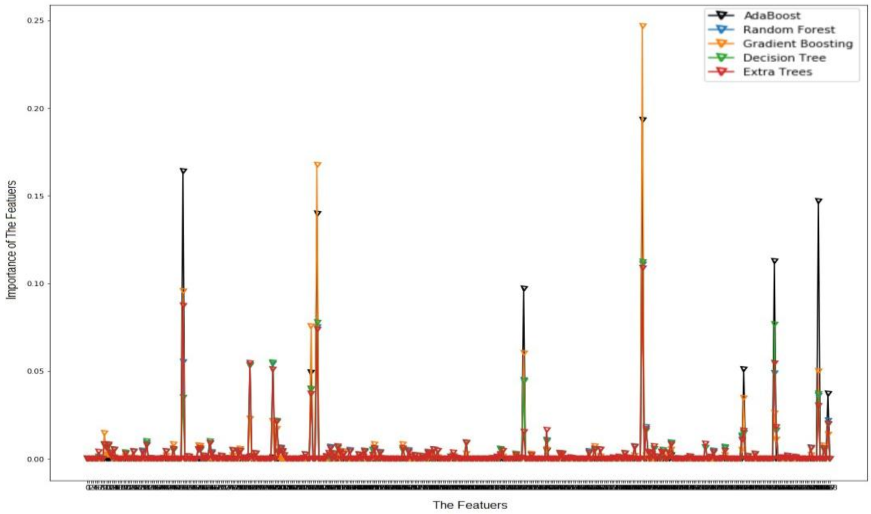

Fig. 2 The x-axis is the 389 features, and the y-axis is the importance rank assigned to each of them

After determining the importance of each feature toward the activities, we chose the ones with a rank higher than 0.004, which were a total of 35 features, to contribute to our classification model.

C. Data Segmentation

Data segmentation takes place in activity recognition after the pre-processing step and the feature selection mechanism. In order for an on-line deep-learning system to effectively classify a particular feature vector representation of the sensor data, the system must include a mechanism for selecting those sensor events from an incoming stream of sensor data. This process is known as segmentation and aims to select the sensor data that lies between activity boundaries. It identifies segments that contain information about the activities to be recognized. Many segmentation methods based on the windowing of streaming data have been used in studies on activity classification, such as time- and sensor-based windowing [25]. In our work, we applied time-based windowing with fixed-sized and overlapping windows to take the values of sensors observed in the previous window. This is because it is generally assumed that due to the higher number of data points, overlapping sliding windows increase the performance of the HAR classifiers when compared to non-overlapping ones [26]. In the experiment, we evaluated the effect of varied window sizes and different overlapping sizes on the recognition results of the proposed system. We got the best results when we set the time period of the window to be 80 with overlapping started after a 5-step distance. This led to 89885 and 38484 samples for the training and testing sets, respectively. Each sample usually contained data associated with multiple labels. We chose for majority labelling, where the label of each segment was specified as the most frequent label in those of the timestamps.

D. Experimental Environment

We implemented our deep-learning model with Google Colaboratory (also known as Colab), which is a cloud service based on Jupyter Notebooks. We used Python runtimes pre-configured with the essential libraries, such as TensorFlow and Keras to build and train our neural networks. The model training and classification were run on an Nvidia Tesla K80 GPU with 2496 CUDA cores and 12 GB RAM. The neural networks were trained in a supervised way by backpropagating the gradients from the softmax layer through to the convolutional layers. The categorical cross-entropy used was the loss function. For optimization, we used Adam [27] with a learning rate of 0.0001. The training was done in 89894k samples and the testing on 38475k samples with a batch size equal to 1024. The hyper-parameters of our proposed deep-learning model for the ContextAct@A4H dataset are provided in Table II.

Table II Relevant Parameters of Each Layer in the Deep 1D-CNN Architecture

Num.	**Components**	**Assigned Parameters**
Layer 1	1dConv	64*2
	Batch Normalization	-
	Activation	ReLU
	Dropout	0.2
Layer 2 to Layer 7	1dConv	64*2
	Batch Normalization	-
	Activation	ReLU
	MaxPool	2
	Dropout	0.2
Layer 8	Flatten Layer	-
Layer 9	Softmax Layer	10 neurons (# of classes)

VI.RESULTS

The results of the performance of our model are presented in this section. In addition, to show the precious of our proposed model, we compared it with baseline results obtained from two deep models.

A. Performance Measurement

Classification accuracy is not an appropriate measure of performance, since activity datasets collected in natural scenes are often imbalanced between classes. In ContextAct@A4H, some classes (e.g., those for sleeping and leaving home) contained a large number of examples while other classes had only a few. Therefore, we evaluated the models using other metrics that were suitable for evaluating class-imbalanced problems. Those metrics were precision, recall, and F1 score, which already have been used in previous works such as [17]. The precision determines the ratio of the relevant points that have been chosen by the model to the total selected points, as shown in the following, where TP is the number of true positives, and FP is the number of false positives:

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (5)$$

The recall is the ratio of relevant points that have been selected by the model to the overall total of the relevant points, as show in the following, where TP is the number of true positives, and FN is the number of false negatives:

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (6)$$

The F1 score is defined in terms of both precision and recall as follows:

$$\text{F1} = 2 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \qquad (7)$$

DOI 10.17148/IJARCCE.2020.9301

B. Experimental Results

The classification results of the proposed method on theContextAct@A4H dataset are presented in this section. We calculated the metrics described previously, which are precision, recall, and F1 score. We also plotted the accuracy and loss curves during the training process for deep 1D-CNN. When observing Table III, the results of precision, recall and F1 score indicate that applying 1D-CNNs for the time-series classification was effective due to their ability to extract very informative deep features from each sequence input to detect the ADLs.

Table III Classification Results of Our Proposed Model (Deep 1D-CNN)

Results of deep 1D-CNN	
Precision	0.92
Recall	0.89
F1 score	0.90
Time Per Epoch	4 sec

On the other hand, according to Fig. 3, the training and testing loss of our model kept decreasing until the last epoch, where both remained in the same range (with a slight decrease in the training loss) when we stopped the training process (training loss = 0.21, validation loss = 0.28). Moreover, the testing accuracy improved until it reached 0.90, while the training accuracy was 0.92. Both then stayed in the same range with no more increases or decreases.

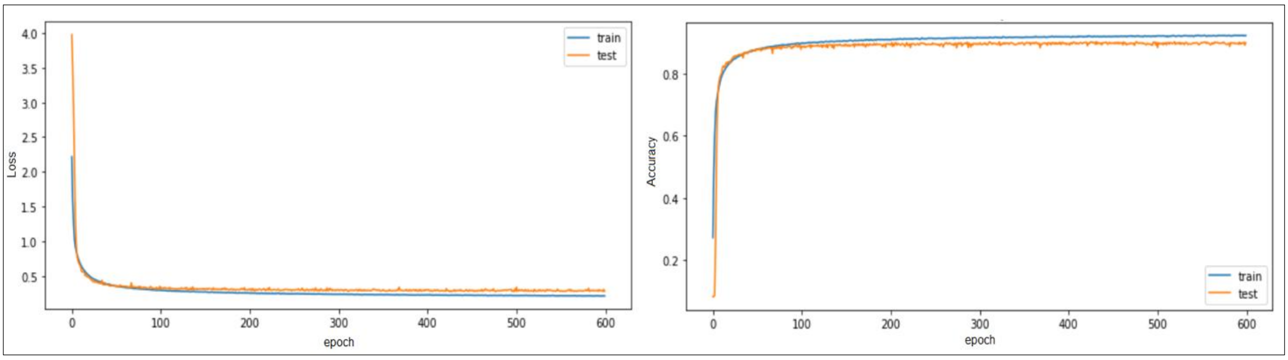

Fig. 3 The left side shows the loss curves while the accuracy curves are at the right side.

The two graphs demonstrate that our model did not suffer from an overfitting problem due to add dropout layers in each convolutional layer which provide regularization and hence prevent overfitting. To support our observation, we trained the model without the dropout layers to see how the training process would progress. We observed that the model without the dropout layers started overfitting the data from the beginning, and we had to stop the training at Epoch 30 where the loss function values for the training and testing data were 0.02 and 0.82, respectively. In addition, the gap between the training and testing accuracy got bigger which also indicates that the model overfitted the data. Finally, the results of training our model with BN provided a lower loss function value than the model without BN for both the training and testing data and also resulted in a higher model accuracy rate.

C. Comparative Results on Deep 1D-CNN with Baseline Models

Table IV represents the results of our model obtained on the ContextAct@A4H dataset when compared to the baseline models using the early mentioned metrics. As mentioned previously, ContextAct@A4H is a newly collected dataset, we could not find any work for activity recognition based deep learning on this dataset to compare our results with. Thus, we compare our network with two baseline models for time series classification in order to illustrate the advantages of our proposed deep 1d-CNN. Those two models are FCN [28] and lstm_128_128_dence [29].

Table IV Performance Comparison with Baseline Models

Model	**Precision**	**Recall**	**F1**	**Time per/epoch**
FCN [28]	0.83	0.81	0.82	1:56m
Lstm_128_128_dence [29]	0.81	0.81	0.81	5:48m
deep 1D-CNN (Our Model)	0.92	0.89	0.90	4s

From Table IV, we can see that our model on the ContextAct@A4H dataset achieved the best overall performance in terms of all the metrics compared to the baseline models where both achieved similar performance with slightly better performance for FCN. The results show that our 1dCNN is superior to the baseline models due to the very deep architecture that utilized one-dimensional convolutions followed by pooling size technique which can automatically and effectively

DOI 10.17148/IJARCCE.2020.9301

derive relevant and discriminative features from a fixed-length segment of the overall dataset to detect ADLs. Even though the recurrent-neural-network-based models (i.e., LSTM) were great techniques for time-series data, since they can find dynamic temporal dependency in a time sequence, in this work, we showed that 1D-CNN can perform better in learning how to classify time series data and successfully recognizing ADLs. In general, recurrent networks usually are preferred in predicting what comes next in a sequence, while CNNs are good at classifying a sequence by extracting very deep features. The results also indicate that our proposed model is faster in learning speeds due to the max-pooling technique, that minimizes the amount of output parameters in each layer which then minimize the computation in the network.

VII. CONCLUSION

In this work, one-dimensional deep convolutional layers equipped with max pooling technique were proposed to deal with time-series data to detect ADLs in ambient intelligent environments. Deep 1D-CNN unifies feature extraction and classification; it can automatically discover and extract the internal structure of the input time-series data and learn deep features to classify ADLs. We evaluated our proposed method on a new real-world dataset, ContextAct@A4H. The results demonstrated that 1D-CNN is one of the most competitive candidates for time-series classification, owing to its ability for learning more robust deep features. Our deep 1D-CNNs outperform other deep-learning-based models (e.g., LSTM) for recognizing ADLs in ambient intelligent environments. In the future, we aim to study the performance of our deep leaning network on other public dataset for ADLs. We also intend to study the effectiveness of recognition ADLs for energy-saving potential in smart homes.

REFERENCES

[1]. J.Jiang, R.Pozza, K.Gunnarsdóttir, N.Gilbert, & K.Moessner, “Using sensors to study home activities,” Journal of Sensor & Actuator Networks, vol. 6, no. 4, p. 32, 2017.
[2]. X. Guo, Z. Shen, Y. Zhang, and T. Wu, “Review on the application of artificial intelligence in smart homes,” Smart Cities, vol. 2, no. 3, pp. 402-420, 2019.
[3]. J.Wang, Y.Chen, S.Hao, X.Peng, & L.Hu, “Deep learning for sensor-based activity recognition: A survey,” Pattern Recognition Letter, vol.119, no.3, 2019.
[4]. D. Sigurdson, and E. Stroulia, “Activity recognition for smart-lighting automation at home,” 9th International Conference on Information, Intelligence, Systems and Applications (IISA), IEEE, 2018, pp. 1-6, 2018.
[5]. W.S. Lima, E. Souto, T. Rocha, R. W. Pazzi, and F. Pramudianto, “User activity recognition for energy saving in smart home environment,” IEEE Symposium on Computers and Communication (ISCC), IEEE, pp. 751-757, 2015.
[6]. F. Richardson, D. Reynolds, and N. Dehak, “Deep neural network approaches to speaker and language recognition,” IEEE Signal Processing Letters, vol. 22, no. 10, pp. 1671-1675, 2015.
[7]. C. Pelletier, G. I. Webb, and F. Petitjean, “Temporal convolutional neural network for the classification of satellite image time series,” Remote Sensing, vol. 11, no. 5, pp. 523, 2019.
[8]. S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” International Conference on Engineering and Technology (ICET), IEEE, pp. 1-6, 2017.
[9]. L. Medsker, and L. C. Jain, Recurrent Neural Networks: Design and Applications, CRC press, 1999.
[10]. M. A. Keyvanrad, and M. M. Homayounpour, “A brief survey on deep belief networks and introducing a new object oriented toolbox (DeeBNet), arXiv preprint arXiv:1408.3264, 2014.
[11]. N.Oukrich, “Daily human activity recognition in smart home based on feature selection, neural network & load signature of appliance, dr. dissertation, 2019
[12]. H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, “Deep learning for time series classification: A review,” Data Mining and Knowledge Discovery, vol. 33, no. 4, pp. 917-963, 2019.
[13]. P. Lago, F. Lang, C. Roncancio, C. Jiménez-Guarín, R. Mateescu, and N. Bonnefond, “The ContextAct@A4H real-life dataset of daily-living activities,” in International and Interdisciplinary Conference on Modeling and Using Context, Springer, pp. 175-188, 2017.
[14]. F.Ordóñez, & D.Roggen, “Deep convolutional & LSTM recurrent neural network for multimodal wearable activity recogn,” Sensor, vol.16, no.1, p.115, 2016
[15]. W. S. Lima, E. Souto, T. Rocha, R. W. Pazzi, and F. Pramudianto, “User activity recognition for energy saving in smart home environment,” IEEE Symposium on Computers and Communication (ISCC), IEEE, pp. 751-757, 2015.
[16]. D. Singh, E. Merdivan, S. Hanke, J. Kropf, M. Geist, and A. Holzinger, “Convolutional and recurrent neural networks for activity recognition in smart environment,” In Towards Integrative Machine Learning and Knowledge Extraction, Springer, pp. 194-205, 2017.
[17]. T. Alshammari, N. Alshammari, M. Sedky, and C. Howard, “Evaluating machine learning techniques for activity classification in smart home environments,” Int. J. Comput. Electr. Autom. Control Inf. Eng, vol. 12, pp. 48-54, 2018
[18]. D. Arifoglu, and A. Bouchachia, “Activity recognition and abnormal behaviour detection with recurrent neural networks,” Procedia Computer Science, vol. 110, pp. 86-93, 2017.
[19]. D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger, “Human activity recognition using recurrent neural networks,” in International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Springer, pp. 267-274, 2017.
[20]. J. Park, K. Jang, and S. B. Yang, “Deep neural networks for activity recognition with multi-sensor data in a smart home,” IEEE 4th World Forum on Internet of Things (WF-IoT), IEEE, pp. 155-160, 2018.
[21]. X. Fan, H. Zhang, C. Leung, and C. Miao, “Comparative study of machine learning algorithms for activity recognition with data sequence in home-like environment,” IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), IEEE, pp. 168-173, 2016.
[22]. H. D. Mehr, H. Polat, and A. Cetin, “Resident activity recognition in smart homes by using artificial neural networks,” 4th International Istanbul Smart Grid Congress and Fair (ICSG), IEEE, pp. 1-5, 2016.
[23]. S.Ioffe, & C.Szegedy, “Batch normalization: Accelerating deep network train by reducing internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015.
[24]. R. Wang, “AdaBoost for feature selection, classification and its relation with SVM, a review,” Physics Procedia, vol. 25, pp. 800-807, 2012.
[25]. N. Yala, B. Fergani, and A. Fleury, “Feature extraction for human activity recognition on streaming data,” International Symposium on Innovations in Intelligent Systems and Applications (INISTA), IEEE, pp. 1-6, 2015
[26]. A. Dehghani, O. Sarbishei, T. Glatard, and E. Shihab, “A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors,” Sensors, vol. 19, no. 22, p. 5026, 2019.
[27]. D. P. Kingma, and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.
[28]. Z. Wang, W. Yan, & T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” International joint conference on neural networks (IJCNN), IEEE, pp. 1578-1585, 2017.
[29]. D.Smirnov, E.M.Nguifo, “Time Series Classification with Recurrent Neural Networks”. In: Advanced Analytics & Learning on Temporal Data, p. 8, 2018.