Enhanced Prediction Model for Human Activity Using an End-to-End Approach

Sangjun Park, Hyung Ok Lee, Yu Min Hwang, *Member, IEEE*, Seok-Kap Ko, and Byung-Tak Lee

***Abstract*—Human activity recognition (HAR) based on ambient sensors aims to recognize a conducted activity. A large number of deep learning models (DLMs) for HAR have been proposed. In contrast, human activity prediction (HAP) aims to early predict an activity. Compared to HAR, the advantage of HAP is to prevent a person from being exposed to unexpected cases by early predicting the activity. However, few DLMs for HAP have been proposed. They predict the next activity via a nonend-to-end fashion, e.g., they take a sequence of consecutive activities where the activities were classified from the sensor information. Thus, the information that which sensors are activated is not used in the prediction. In this study, we propose an end-to-end HAP model to predict the next activity from a sequence of consecutive events. The model has an encoder, a classifier, and a regressor. The encoder gives an encoded vector by encoding events. The regressor learns temporal dependency from a sequence of encoded vectors to predict the next encoded vector. The classifier predicts the next activity using the next encoded vector. We use the Milan and Aruba data sets to study a prediction accuracy of the model. We compare our model with a nonend-to-end model based on long-term memory, taking a sequence of past activities. We show that our model achieves the better prediction accuracy than the nonend-to-end model by up to 4.73% and 7.39% for Milan and Aruba, respectively, meaning that the information related to events can be used in the prediction.**

***Index Terms*—Human activity prediction (HAP), human activity recognition (HAR).**

I. Introduction

With the advent of medical technologies, people can live independently in their homes for a longer time, leading to the growth in the number of elderly people [1]. As the elderly people tend to live alone, it is a matter of great concern to make systems, such as healthcare and monitoring services to improve the quality of life or prevent diseases [2], [3]. These systems, for example, can be used to monitor the elderly for aiding care or assisting impaired elderly. To this end, they have to take the ability to recognize what an elderly person conducts. For this end, the study of human activity recognition (HAR) [46], [47] has received significant attention in the literature for decades.

Then, another capability of the aforementioned systems is the ability to predict sequential activities of the elderly in the near future. Specifically, the systems not only recognize a currently conducted activity but also predict an activity expected to occur in the near future. If the systems can predict future activities, a proper service can be prepared. Let us assume, for example, that an elderly person is currently sleeping. If we predict that he/she awakes within 5 mins, we can make a service to heat up coffee or tea. Also, an elderly person can often suffer from a dementia problem such as Alzheimer's disease. If the predicted activities can be often different to the activities actually conducted by this person, we can send messages to his caretakers to improve the quality of a care service as shown in [19]. Hence, the study of human activity prediction (HAP) is just as important as the study of HAR.

A. Existing Studies Using Deep Learning Models for HAR

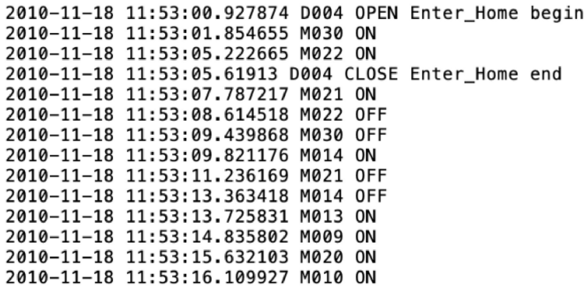
Nowadays, most of the studies [4], [5], [6], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35] on HAR have been conducted using deep learning models (DLMs), such as long short-term memory (LSTM) and convolutional neural network (CNN) when signals obtained using wearable sensors, e.g., smart watches and smart phones, are provided. We provide the summaries of the existing DMLs.

In [4], Ordonez and Roggen proposed a model based on deep CNN (DCNN) and LSTM. The DCNN is used to extract spatial features, and then the LSTM is used to learn temporal relations among these features. A similar model was proposed in [5]. Cho and Yoon [25] proposed a two-steps model based on CNN. First, a sample is determined to be either a static group or a dynamic group. Second, this sample is recognized from a selected group. A similar model was proposed in [26]. Then, Garcia et al. [27] proposed an ensemble model of multiple auto-encoders (AEs). Each AE is trained to learn a single activity. The recognition is performed by selecting the AE that achieves the smallest error. Sena et al. [28] proposed an ensemble model in which it has three DCCNs. Each DCNN has a different kernel size to extract features in a different view. All of the features are vertically stacked to construct a feature that is used for the recognition. Gao et al. [29] extracted a feature using a stacked AE and used lightgbm to recognize an activity from this feature. Besides, to explicitly utilize sensor modalities, an attention module was used with DLMs in [30], [31], [32], [34], and [35]. Al-Qaness et al. [48]

Manuscript received 22 May 2022; revised 28 July 2022 and 16 September 2022; accepted 10 November 2022. Date of publication 21 November 2022; date of current version 24 March 2023. This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korea Government (MSIT, Solution for Social Problems Using Personalized Care Service for Vulnerable Social Groups) under Grant 2020-0-01861. *(Corresponding author: Byung-Tak Lee.)*

The authors are with the Honam Research Center, Electronics and Telecommunications Research Institute, Gwangju 61012, Republic of Korea (e-mail: bytelee@etri.re.kr).

Digital Object Identifier 10.1109/JIOT.2022.3223674



```
2010-11-18 11:53:00.927874 D004 OPEN Enter_Home begin
2010-11-18 11:53:01.854655 M030 ON
2010-11-18 11:53:05.222665 M022 ON
2010-11-18 11:53:05.61913 D004 CLOSE Enter_Home end
2010-11-18 11:53:07.787217 M021 ON
2010-11-18 11:53:08.614518 M022 OFF
2010-11-18 11:53:09.439868 M030 OFF
2010-11-18 11:53:09.821176 M014 ON
2010-11-18 11:53:11.236169 M021 OFF
2010-11-18 11:53:13.363418 M014 OFF
2010-11-18 11:53:13.725831 M013 ON
2010-11-18 11:53:14.835802 M009 ON
2010-11-18 11:53:15.632103 M020 ON
2010-11-18 11:53:16.109927 M010 ON
```

Fig. 1. Example of a data set given by the CASAS project.

proposed a multilevel residual network model with an attention module. In this model, each residual network is used to extract a local feature from data provided by a corresponding inertial measurement unit. A bidirectional gated recurrent unit with the attention module is used to extract a final feature from the local features and this final feature is used for the recognition. Finally, there have been works to develop lightweight models that are expected to be used in low-power edge devices. As an example, Helmi et al. [49] proposed a lightweight feature selection method using the gradient-based optimizer algorithm and used a multiclass SVM for the recognition. Rashid et al. [50] an adaptive CNN model. In this model, there is a module that takes a statistical feature to select a portion of the CNN model for the recognition, making this model energy efficient.

These studies state that an activity recognition can be possible with a high probability using the wearable sensors. However, there is a problem such that an elderly frequently takes off these sensors for a long time. We therefore cannot get signals from the sensors, making the activity recognition impossible.

To deal with this problem, in [7], [8], [9], and [10], ambient sensors, such as motions sensors, door sensors, and temperature sensors are used. Unlike to the wearable sensors, these ambient sensors are unobtrusively placed in a home to get the information, such as temperatures and movements. For example, we consider Fig. 1. This figure shows an example in [21] where the ambient sensors were used. It is noted that strings with *M* and *D* are denoted as motion and door sensors, respectively. Each row has the status of a sensors represented as a form of binary event. For example, we consider the 2nd row and the 7th row in the example. The 2nd row shows that the motion sensor M030 was activated when 18 November 2010 11:53:01. The 7th row shows that this motion sensor was deactivated. We then provide the following summaries on the existing DLMs for HAR when ambient sensors are used.

Singh et al. [11], [12] proposed models based on LSTM and CNN. They empirically showed that their models achieved better recognition accuracy than machine learning models. Gochoo et al. [13] proposed a three-stages framework such that: 1) they segmented a data set to get activity images; 2) they used CNNs to extract a feature of each activity image; and 3) they used a fully connected (FC) layer for the recognition. Tan et al. [14] extended this framework to a case where residents lived in a single home. Medina-Quero et al. [15] developed a feature extraction method using a fuzzy rule and proposed an ensemble model for the recognition. Hamad et al. [16] extended the work in [15] by exploiting oncoming events. Machot et al. [17] proposed a feature extraction method based on a random forest method. Then, they proposed a model based on recurrent neural network (RNN) for the recognition. Bouchabou et al. [18] proposed a feature extraction method based on techniques in the natural language processing domain. Then, they proposed a model based on CNN to recognize an activity. Wang et al. [36] used a stacked AE to extract a feature and used an FC layer for the recognition. Wang et al. [37] proposed two encoding schemes in which they encoded a segmented data into either a vector or a matrix. They then proposed a model based on CNN, LSTM, and stacked AE for the recognition. Hamad et al. [38] proposed a model based on dilated casual CNNs with a self-attention module.

B. Existing Studies Using Deep Learning Models for HAP

Vision sensors such as camera can be used to obtain video frames in which each frame records silhouettes of an elderly. From a sequence of consecutive video frames, studies on HAP were conducted in [39], [40], [41], and [42]. However, the usage of the vision sensors can lead to a privacy problem such that we can directly guess a current activity of an elderly from the silhouettes. This can be critical for some elderly who prefer to their privacy as reported in [44] and [45].

In contrast, as we have illustrated in Fig. 1, we consider the data set obtained using the ambient sensors. From this data set, we can only know the trajectory of an elderly, implying that it is hard for us to directly guess a current activity from the data set. We therefore can prevent the privacy problem from arising by considering the usage of the ambient sensors.

To the best of our knowledge, only few studies on HAP have been conducted using DLMs when the ambient sensors are used. For example, in [19], Tax proposed a model based on LSTM to predict the next activity and its timestamp. Krishna et al. [20] proposed a model based on LSTM to estimate probabilities for the next activity and its duration. The results of these studies showed that the models achieved better performance when they were compared with sequence prediction methods.

We consider a case where we apply the model in [19] into the example in Fig. 1. Since this example has a sequence of binary events, a classifier is used to infer activities from the sequence. The model extracts a feature from the inferred activities. Then, the model predicts the next activity using this extracted feature. The model thus can be considered to be a nonend-to-end model.

However, there are three issues on the model. We will provide details in Section II-B. Here, we shortly introduce them as follows.

1) A classifier can yield misrecognized activities.
2) Sensor data is not used in the prediction.
3) Human knowledges can be required to extract features.

These issues can be also observed when we consider the model in [20].

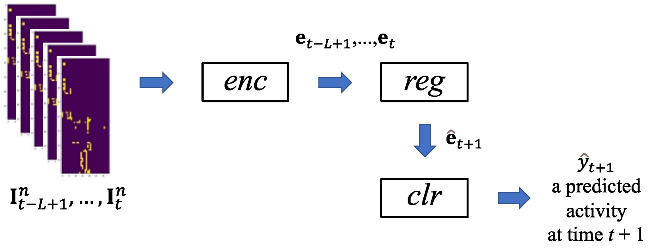

$\mathbf{I}^n_{t-L+1}, \ldots, \mathbf{I}^n_t$
enc
$\mathbf{e}_{t-L+1}, \ldots, \mathbf{e}_t$
reg
$\hat{\mathbf{e}}_{t+1}$
clr
$\hat{y}_{t+1}$
a predicted
activity
at time $t+1$

Fig. 2. Architecture of the proposed end-to-end model for HAP.

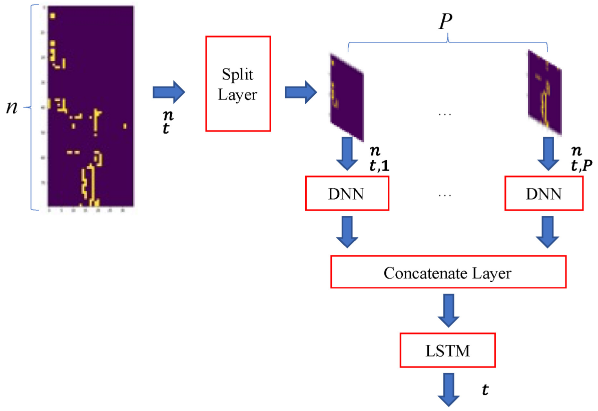


Fig. 3. Architecture of the encoder.

C. Motivations, Novelty, and Contributions

The motivation of this article is to propose a model which can address the issues of the models in [19] and [20]. To this end, we propose an end-to-end model shown in Fig. 2. This model takes[1] the sensor data to conduct the prediction, which implies that any decisions to infer activities from the sensor data are not required. Besides, as the model takes the sensor data, we expect that a better prediction performance can be achieved compared to the nonend-to-end model. This is due to two reasons as follows: 1) the sensor data given by motion sensors can include the information on the trajectory of an elderly and 2) particular sensors can be involved in some activities, e.g., a sensor nearby by a bed can be highly related to a sleeping activity. Finally, the model automatically extracts features without human knowledges as we train the model via an end-to-end fashion.

To show how our model conducts the prediction, we give an example where a sequence of binary events from time $n-1$ to time t is provided. First, we split the sequence into two subsequences. The first one has the binary events from time $n-1$ to time $t-1$ while the second one has those from time n to time t. Second, we use an encoder to encode each one. For simplicity, the encoded vector of the first one is denoted as $\mathbf{e}_t$, which will be defined in (5). The remain one is denoted as $\mathbf{e}_{t-1}$. Since the vectors are originated from the overlapped subsequences, they have temporal dependency. We third use this dependency by defining a regressor that predicts $\widehat{\mathbf{e}}_{t+1}$ from $\mathbf{e}_t$ and $\mathbf{e}_{t-1}$. We finally use a classifier that takes $\widehat{\mathbf{e}}_{t+1}$ to predict an activity at time $t+1$.

The novelty in our model is the above encoder. We use this encoder to get a sequence of encoded vectors from the sensor data gathered until time t. We then use the regressor to obtain an encoded vector at time $t+1$ using the sequence given by the encoder. Thus, the encoder has to take an ability to extract temporal and discriminative features from the sensor data. To this end, we implement the encoder using multiple DNNs and a single LSTM where the DNNs are used to encoded the data and the LSTM is used to capture the temporal dependency on the encoded data. The details on the encoder illustrated in Fig. 3 will be given in Section III.

The main contribution in this manuscript is that we propose the first end-to-end model that predicts future activities from a sequence of binary events. This model uses the encoder to overcome the limitations of the models in [19] and [20] stated in the previous section. For example, to use these models, there is a prestage such that we first use a classifier to get a sequence of activities from a sequence of binary events. In contrast, our model does not require this prestage. Also, the encoder in our model extracts the temporal and discriminative features of the binary events, meaning that our model can uses the sensor data for the prediction. Finally, by training our model via an end-to-end style, our encoder is automatically learned to extract features. To sum up, the contributions of can be summarized as follows.

1) We propose the first end-to-end model to predict future activities from a given sequence of binary events without recognizing past activities.
2) We use two data sets, such as Aruba [21] and Milan [22] to compare the proposed model with a nonend-to-end model based on LSTM, taking a sequence of recognized activities. As a result, we show that the proposed model outperforms the nonend-to-end model with respect to a prediction accuracy. The proposed model can achieve a better prediction accuracy of up to 4.73% and 7.39% for Milan and Aruba, respectively.
3) We conduct hypothesis tests to confirm that our end-to-end model can outperform the nonend-to-end model in terms of the prediction accuracy.
4) We provide intuitive explanations and simulation results to explain that the proposed model can outperform the nonend-to-end model.

Finally, we organize the remainder of this article as follows. In Section II, we provide the notations, preprocessing, and task formulation. In Section III, we introduce our end-to-end model for HAP. We demonstrate the experimental results and their interpretations in Section IV. In Section V, the conclusions are presented.

II. Notations, Preprocessing, and Task Formulation

A. Notations and Preprocessing

The following notation is used throughout this article: a small bold letter $\mathbf{f}$ represents a vector, whereas a capital bold letter $\mathbf{F}$ represents a matrix. Consider a model f, its parameter is denoted as θ_f. The superscript T denotes the transpose operation.

[1] The proposed model takes a sequence of binary images generated from the sensor data where each image is formed according to (1).

A data set has a sequence of binary events as we have shown in Fig. 1. Hence, a preprocessing is required to make an input. To this end, we begin to divide this data set into equal time slices of a fixed value Δ. Let $x_t^s \in \{0, 1\}$ be the sth sensor data at time t, indicating whether the sth sensor is activated from time t to time $t + \Delta$ at least once. If the sth sensor is activated, x_t^s is set to 1. Otherwise, x_t^s is set to 0. If there are S sensors, we have x_t^s where $t = 0, 1, \dots, T$ and $s = 1, 2, \dots, S$. We then define a vector $\mathbf{x}_t$ to represent all the sensor data at time t as follows:

$$\mathbf{x}_t := \begin{bmatrix} x_t^1 & x_t^2 & \cdots & x_t^S \end{bmatrix}.$$

The label of $\mathbf{x}_t$ is denoted as $y_t \in \{0, \dots, C - 1\}$, where C is the number of activities. Note that if $\mathbf{x}_t$ has no label, we exclude it.

Given a sequence of n consecutive sensor data, $\mathbf{x}_{t-n+1}, \dots, \mathbf{x}_t$ at time t, we can form a binary image $\mathbf{I}_t^n$ of size $n \times S$

$$\mathbf{I}_t^n \triangleq \begin{bmatrix} \mathbf{x}_{t-n+1}^{\mathrm{T}} & \cdots & \mathbf{x}_t^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{1}$$

where the t-th row vector of $\mathbf{I}_t^n$ corresponds to $\mathbf{x}_i$. We then set the label of this binary image to y_t.

B. Task Formulation

We formulate a task for HAP using the above-defined symbols as follows.

HAP–Task: The aim of this task is to construct a model f that *predicts* an activity at time $t + 1$ from a sequence of L binary images $\mathbf{I}_{t-L+1}^n, \dots, \mathbf{I}_t^n$

$$f : \mathbf{I}_{t-L+1}^n, \dots, \mathbf{I}_t^n \to \hat{y}_{t+1}. \tag{2}$$

We note that the task considered in [19] and [20] can differ from our task. Their task is to propose a model f below

$$f : h(\mathbf{a}_{t-L+1}, \dots, \mathbf{a}_t) \to \hat{y}_{t+1} \tag{3}$$

where $\mathbf{a}_t$ is an activity at time t and $h(x)$ is a feature of x. We then give three comments on the model in (3) as follows.

First, Tax [19] and Krishna et al. [20] assumed that a sequence of past activities is given. However, to get this sequence, we use a classifier for recognizing past activities. Then, as we have stated in Section I-B, the classifier can yield misclassified ones. Hence, a prediction performance of the model can be degraded no matter how the model is well trained.

Second, there is a chance to improve a prediction performance by utilizing the sensor data. We have stated that the sensor data can be used to show the movements of an elderly. For example, from Fig. 1, we can see a trajectory representing that an elderly moved from a place equipped with the M030 sensor to a place equipped with the M010 sensor. Hence, a probability that a next activity is one of activities related the place equipped with the sensor M010 is high. We also invoke a heat map generated in [13]. This map shows a distribution of sensors for each activity. For example, from the map, it can be seen that the M026 sensor and the M027 sensor were mainly involved in the Work activity. Thus, the sensor data should be utilized in the prediction.

Third, human knowledges are required to extract a feature that is taken by the model. For example, Tax [19] used the following knowledges such that "activities can be correlated to the point in time at which they occur."

On the other hand, the model in (2) is trained using a sequence of consecutive binary images without the usage of any classifier. Also, we remind that each binary image has the information to indicate which sensors are activated. Finally, the model is trained via an end-to-end fashion, making that the model can be capable for extracting discriminative features without any knowledges.

III. Methodology

In this section, we propose an end-to-end model for HAP. We illustrate the architecture of this model in Fig. 2 and provide the following summaries.

The HAP model has an encoder, a regressor, and a classifier, denoted as *enc*, *reg*, and *clr*, respectively. At time t, the encoder takes each $\mathbf{I}_t^n$ defined in (1) to get an encoded vector denoted as $\mathbf{e}_t$ that will be defined in (4). The regressor takes a sequence of consecutive encoded vectors from time $t - L + 1$ to time t to predict an encoded vector at time $t + 1$. The classifier then takes this predicted vector to recognize an activity at time $t + 1$.

A. End-to-End HAR Model

The architecture of the encoder is illustrated in Fig. 3. Given a binary image $\mathbf{I}_t^n$, we split it into P subimages $\mathbf{I}_{t,1}^n, \dots, \mathbf{I}_{t,P}^n$ in which the size of each one is defined in advance. Once the subimages are prepared, we conduct an encoding routine to encode each subimage as follows.

Let $\mathbf{c}_i^j$ be the jth column vector of the ith subimage. Then for each column, we use the ith deep neural network (DNN) f_i to encode $\mathbf{c}_i^j$ into a real value as follows:

$$c_i^j \triangleq f_i\left(\mathbf{c}_i^j; \theta_i\right) \in \mathcal{R}$$

where $j = 1, \dots, S$, and $i = 1, \dots, P$. We note that each DNN has FC layers in which the activation function of each layer is *ReLU*. Then, the number of output units of the last layer is set to be 1 to get a real value. In Section IV-B, we will give details on hyperparameters of the DNNs.

By gathering these S values, we form a vector of size $1 \times S$. By applying this routine to each subimage, we get a sequence of P vectors. We then form a matrix of size $P \times S$ by vertically stacking the sequence and use LSTM to encode this matrix for yielding an encoded vector. A classifier then takes this encoded vector to recognize an activity.

To train both of the encoder and the classifier, we minimize the following loss function:

$$L\left(\theta_f\right) \triangleq \mathrm{CE}\left(\mathrm{clr}\left(\mathrm{enc}\left(\mathbf{I}_t^n; \theta_{\mathrm{enc}}\right); \theta_{\mathrm{clr}}\right), y_t\right) \tag{4}$$

where $\mathrm{CE}(x, y)$ denotes the cross-entropy between x and y and $\theta_f \triangleq \{\theta_{\mathrm{enc}}, \theta_{\mathrm{clr}}\}$ is a trainable parameter.

We provide three comments regarding the encoder as follows: First, given a subimage $\mathbf{I}_{t,i}^n$, we use the ith DNN to encode each column of $\mathbf{I}_{t,i}^n$. That is, we follow the same rule to encode $\mathbf{I}^n_{t,i}$. Second, we consider two subimages, such as $\mathbf{I}^n_{t,1}$ and $\mathbf{I}^n_{t,P}$. Even if these subimages come from $\mathbf{I}^n_t$, they have different information. That is, the first subimage has past information while the last one contains recent information. We thus intend to use multiple DNNs to encode each subimage for exploiting the diversity among the subimages. Finally, we can also use a single DNN to encode all of the subimages. However, this makes that the HAR model cannot use the diversity among the subimages. This may degrade the recognition performance of the HAR model.

During the review process, we find that the model in [48] is similar to our HAR model. As we have stated in Section I-A, this model extracts a local feature from the corresponding sensor's data. Then, it constructs a final feature from these local features by applying a bidirectional gated recurrent unit with an attention module. Similarly, the encoder in our model extracts a set of local features and uses LSTM to form a feature from these local features. However, in our model, all the sensor data are used to form the local features. In contrast, in [48], the data of each sensor is used to form a corresponding local feature.

B. Proposed End-to-End HAP Model

We assume that both of the encoder and the classifier has been trained. That is, the encoder can extract the encoded vector of a binary image $\mathbf{I}^n_t$. For simplicity, let this vector at time t be

$$\mathbf{e}_t \triangleq \text{enc}\big(\mathbf{I}^n_t; \theta_{\text{enc}}\big) \tag{5}$$

where θ_{enc} is a nontrainable parameter.

Given consecutive images, such as $\mathbf{I}^n_t$ and $\mathbf{I}^n_{t-1}$, the images can share the sensor data gathered from time $t - n + 1$ to time $t - 1$. Thus, $\mathbf{e}_t$ and $\mathbf{e}_{t-1}$ can have this dependency, which can be used to predict $\mathbf{e}_{t+1}$. To this end, we use a regressor to estimate the encoded vector at time $t + 1$ as follows:

$$\widehat{\mathbf{e}}_{t+1} \triangleq \text{reg}\big(\mathbf{e}_{t-L+1}, \ldots, \mathbf{e}_t; \theta_{\text{reg}}\big) \tag{6}$$

which takes a sequence of the L consecutive encoded vectors from time $t - L + 1$ to time t. The regressor is trained to learn the dependency intrinsic in the sequence by minimizing the loss function as follows:

$$L\big(\theta_{\text{reg}}\big) \triangleq \text{MSE}\big(\text{reg}\big(\mathbf{e}_{t-L+1}, \ldots, \mathbf{e}_t; \theta_{\text{reg}}\big), \mathbf{e}_{t+1}\big) \tag{7}$$

where $\text{MSE}(x, y)$ is the mean-squared error between x and y. We use this model to predict the activity at time $t + 1$ as follows:

$$\hat{y}_{t+1} \triangleq \text{clr}\big(\text{reg}\big(\mathbf{e}_{t-L+1}, \ldots, \mathbf{e}_t; \theta_{\text{reg}}\big); \theta_{\text{clr}}\big).$$

To predict the activity at time $t + 2$, we again use the model as follows:

$$\hat{y}_{t+2} \triangleq \text{clr}\big(\text{reg}\big(\mathbf{e}_{t-L}, \ldots, \widehat{\mathbf{e}}_{t+1}; \theta_{\text{reg}}\big); \theta_{\text{clr}}\big).$$

This routine is repeated until we predict activities from time $t + 1$ to time $t + F$.

In summary, we train this model via a transfer learning. That is, we train both of the encoder and the classifier by minimizing the loss function defined in (4), and freeze their parameters. We then use this trained encoder to train the regressor by minimizing the loss function defined in (7). By doing so, we only update the parameters of the regressor because those of the encoder are fixed. We last concatenate the encoder, the regressor and the classifier to build the model shown in Fig. 2.

We note that there is a case in which a predicted vector, e.g., $\widehat{\mathbf{e}}_{t+1}$, can be distorted. In this case, there is a possibility that the classifier can provide an incorrect decision. To prevent such a case, we can jointly train both the regressor and the classifier by minimizing a loss function defined as follows:

$$\begin{aligned} L\big(\theta_{\text{clr}}, \theta_{\text{reg}}\big) \triangleq\ & \text{MSE}\big(\text{reg}\big(\mathbf{e}_{t-L+1}, \ldots, \mathbf{e}_t; \theta_{\text{reg}}\big), \mathbf{e}_{t+1}\big) \\ & + \lambda \text{CE}(\text{clr}(\widehat{\mathbf{e}}_{t+1}; \theta_{\text{clr}}), y_{t+1}) \end{aligned}$$

where the initial parameters of θ_{clr} are taken from the trained classifier, and λ is a nonnegative value.

IV. Experiments

We study the prediction performance of the proposed model using two data sets, Aruba [21] and Milan [22] by comparing the proposed model with a nonend-to-end model that predicts the next activity from past activities.

A. Data Sets

We use two public data sets named as Aruba and Milan for our experiments. The Aruba data set [21] contains sensor data where a woman conducted ten activities. This data was collected from 4 November 2010 to 11 June 2011 using five temperature sensors, four door sensors, and 31 motion sensors. The Milan data set [22] contains sensor data in a home where a woman with a pet conducted 15 activities. This data was collected from 16 October 2009 to 6 January 2010 using three door sensors, two temperature sensors, and 28 motion sensors.

Both of the data sets are all represented as a sequence of sensor data, as we have shown in Fig. 1. We exclude all the sensor data of the temperature sensors. This makes that these data sets only have a sequence of binary events in which each event shows the status of a corresponding sensors. Then, as we have stated in Section II, we divided these data sets into equal time slices of 1 min to form samples. The results are given in Table I and II for Aruba and Milan, respectively.

B. Hyper Parameters

We explain how we set the parameters for each component in Fig. 3. In the encoder (*enc*), we split a binary image $\mathbf{I}^n_t$ of size $60 \times S$ into six subimages. Except for the 6th subimage, the size of each subimage is $20 \times S$ and that of the 6th subimage is $10 \times S$. Let $\mathbf{I}^{60}_t$ be a sequence of consecutive binary events from time $t - 60$ to time t. Except for the 6th subimage, the ith subimage has a sequence of consecutive binary events from time $t - (7 - i) \times 10$ to time $t - (5 - i) \times 10$, where $i = 1, \ldots, 5$. The 6th subimage has the sequence from time $t - 50$ to time t. For the ith subimage except for the 6th subimage, each DNN consists of three FC layers as follows:

$$f_i \triangleq \text{FC}(16, \text{relu}) \rightarrow \text{FC}(8, \text{relu}) \rightarrow \text{FC}(1, \text{relu})$$

TABLE I
NUMBER OF SAMPLES OF EACH ACTIVITY IN ARUBA [21]

Name of datasets	Aruba [21]
Activity	# of samples
Bed_to_Toilet	592
Eating	2876
Enter_Home	483
Housekeeping	705
Leave_Hoome	474
Meal_Preparation	14125
Relax	99460
Sleeping	62631
Wash_Dishes	529
Work	3088

TABLE II
NUMBER OF SAMPLES OF EACH ACTIVITY IN MILAN [22]

Name of datasets	Milan [22]
Activity	# of samples
Bed_to_Toilet	420
Chores	706
Desk_Activity	798
Dining_Rm_Activity	353
Eve_Meds	29
Guest_Bathroom	1278
Kitchen_Activity	8085
Leave_Home	4159
Master_Bathroom	2251
Master_Bedroom _Activity	2282
Meditate	125
Morning_Meds	87
Read	11260
Sleep	33358
Watch_TV	6036

where $i = 1, \ldots, 5$, and *relu* denotes the *ReLU* activation function. For the 6th subimage, we use the 6th DNN as follows:

$$f_6 \triangleq \mathrm{FC}(8, \mathrm{relu}) \rightarrow \mathrm{FC}(1, \mathrm{relu}).$$

We use LSTM with 20% *dropout* where the number of outputs is 32. The classifier *clr* is defined as follows:

$$\mathrm{clr} \triangleq \mathrm{FC}(C, \mathrm{softmax})$$

where C is the number of activities and *softmax* is the *softmax* activation function. The regressor *reg* is defined as follows:

$$\mathrm{reg} \triangleq \mathrm{LSTM}(32) \rightarrow \mathrm{dropout}(20\%)$$

which takes a sequence of L encoded vectors to predict the next encoded vector of size 32×1.

C. Experiments for HAR

We remind that the regressor takes a sequence of consecutive encoded vectors in which each vector is created using the pretrained encoder. Then, the pretrained classifier takes the output of the regressor to predict the next activity. Thus, the prediction performance of the model depends on how both of the encoder and the classifier is well trained. We run experiments to study a recognition performance of an end-to-end HAR model that we have defined in Section III-A.

To conduct the experiments, we randomly select at most 5000 samples of each activity because the data sets are imbalanced. The batch size is set to be 32 and the epoch is set to be 16. We use the *Adam* optimizer with a learning rate of 0.001 and exploit the fivefold cross-validation method. To evaluate a recognition performance, we use metrics, such as accuracy, recall, precision, and F1-score. They are defined as follows:

$$\begin{aligned}
\mathrm{Accuracy} &= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \\
\mathrm{Recall} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \\
\mathrm{Precision} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \\
\mathrm{F1{-}score} &= \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}
\end{aligned}$$

where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives, respectively.

We show the evaluated performances for the data sets in Tables III and IV, respectively. For Aruba, the average accuracy and the average F1-score were 93.58% and 89.37%, respectively. Then, for Milan, those were 84.58% and 68.95%, respectively. It can be seen that the recognition performance for Milan was not as good as that for Aruba. As an example, the average F1-score for Milan was 20.42% smaller than that for Aruba. The average accuracy for Milan was 9% smaller than that for Aruba.

We give two interpretations regarding the results. First, as we have shown in Tables I and II, Milan is more imbalanced than Aruba. Besides, the number of samples of the activities, such as Eve_Meds and Morning_Meds in Milan is extremely small. As a result, they could not be recognized by the model, as we have shown in Table IV. Second, as we have stated in Section IV-A, there was a woman with a dog in Milan while there was a woman in Aruba. All the events in Aruba were generated by the woman, while some events in Milan could be generated by the dog. These events generated by the dog were noisy and nonuseful, preventing the model from recognizing the activities of Milan.

We prepare Table V for comparing our model with the existing models when the models are tested using the Aruba data set. As we have shown in Table V, the average F1-score of our model was 89.37%, which is better than both of the DCNN model in [13] and the RNN model in [17] by up to 10.37% and 4.34%, respectively. This concludes that the trained encoder can extract discriminative features, and the trained classifier can utilize this feature to recognize an activity.

D. Explanations on Nonend-to-End Model for HAP

We remind the models in [19] and [20]. As we have stated in Section I-B, these models can be used to predict next activities via a nonend-to-end fashion. Namely, they take an input obtained using a classifier in advance. For example, the model in [20] predicts the next activity by taking a sequence of past activities and their durations as an input where each activity was classified in advance. We thus decide to compare our

TABLE III
EVALUATED PERFORMANCES OF OUR HAR MODEL FOR ARUBA [21]

Name of datasets	Aruba [21]			
Activity	Precision	Recall	F1-score	Accuracy
Bed_to_Toilet	92.85%	98.65%	95.66%	98.65%
Eating	91.48%	90.68%	91.08%	90.68%
Enter_Home	84.68%	80.12%	82.34%	80.12%
Housekeeping	89.20%	86.67%	87.91%	86.67%
Leave_Hoome	77.25%	85.23%	81.04%	85.23%
Meal_Preparation	89.25%	92.77%	90.98%	92.77%
Relax	97.04%	95.64%	96.33%	95.64%
Sleeping	99.37%	98.05%	98.71%	98.05%
Wash_Dishes	82.18%	62.76%	71.17%	62.76%
Work	98.01%	98.90%	98.45%	98.90%
Average	90.13%	88.95%	89.37%	93.58%

TABLE IV
EVALUATED PERFORMANCES OF OUR HAR MODEL FOR MILAN [22]

Name of datasets	Milan [22]			
Activity	Precision	Recall	F1-score	Accuracy
Bed_to_Toilet	72.73%	57.14%	64.00%	57.14%
Chores	71.21%	60.62%	65.49%	60.62%
Desk_Activity	84.65%	84.34%	84.49%	84.34%
Dining_Rm_Activity	84.50%	78.75%	81.52%	78.75%
Eve_Meds	00.00%	00.00%	00.00%	00.00%
Guest_Bathroom	64.77%	71.36%	67.91%	71.36%
Kitchen_Activity	80.55%	79.12%	79.83%	79.12%
Leave_Home	95.24%	83.27%	88.85%	83.27%
Master_Bathroom	70.55%	71.61%	71.08%	71.61%
Master_Bedroom _Activity	71.37%	73.75%	72.54%	73.75%
Meditate	73.48%	77.60%	75.49%	77.60%
Morning_Meds	50.00%	03.45%	06.45%	03.45%
Read	91.27%	92.02%	91.64%	92.02%
Sleep	85.39%	98.04%	91.27%	98.04%
Watch_TV	93.89%	93.46%	93.38%	93.46%
Average	72.64%	68.30%	68.95%	84.58%

TABLE V
COMPARISON OF THE PROPOSED MODEL WITH THE EXISTING MODELS IN TERMS OF THE AVERAGE F1-SCORE FOR THE ARUBA DATA SET [21]

Datasets	Model	Average F1-Score
Aruba [21]	The proposed model	**89.37%**
	DCNN [13]	79.00%
	RNN [17]	85.03%

model with a nonend-to-end model corresponding to (3) in which this model is based on LSTM.

To train this nonend-to-end model, we use a sequence of one-hot encoded vectors formed from the ground-truth activities. The model then predicts the next activity from a sequence of one-hot encoded vectors, in which each vector is constructed from an activity recognized by the HAR model used in the previous section.

E. Experiments for HAP

We conduct experiments to show that the proposed model can outperform the nonend-to-end model in terms of a prediction performance.

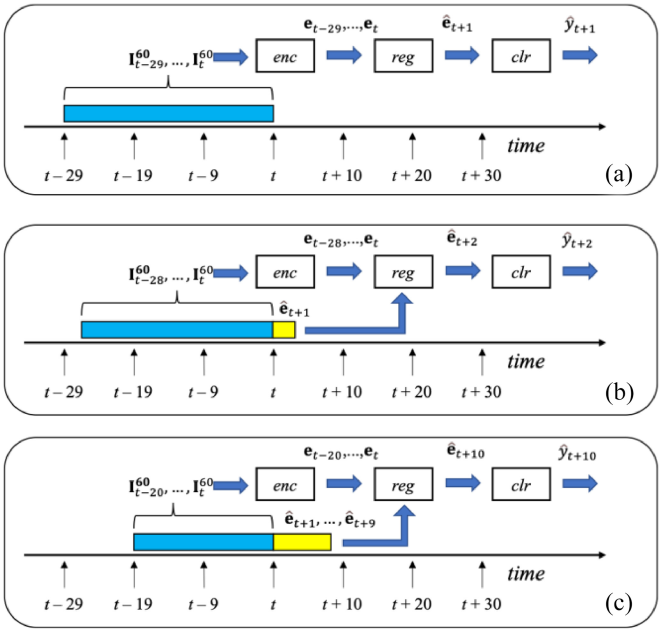

$\mathbf{e}_{t-29},\ldots,\mathbf{e}_t$
$\hat{\mathbf{e}}_{t+1}$
$\hat{y}_{t+1}$
$\mathbf{I}^{60}_{t-29},\ldots,\mathbf{I}^{60}_{t}$
enc
reg
clr
time
$t-29$
$t-19$
$t-9$
t
$t+10$
$t+20$
$t+30$
(a)
$\mathbf{e}_{t-28},\ldots,\mathbf{e}_t$
$\hat{\mathbf{e}}_{t+2}$
$\hat{y}_{t+2}$
$\mathbf{I}^{60}_{t-28},\ldots,\mathbf{I}^{60}_{t}$
$\hat{\mathbf{e}}_{t+1}$
(b)
$\mathbf{e}_{t-20},\ldots,\mathbf{e}_t$
$\hat{\mathbf{e}}_{t+10}$
$\hat{y}_{t+10}$
$\mathbf{I}^{60}_{t-20},\ldots,\mathbf{I}^{60}_{t}$
$\hat{\mathbf{e}}_{t+1},\ldots,\hat{\mathbf{e}}_{t+9}$
(c)

Fig. 4. Illustration of the prediction.

TABLE VI
AVERAGE PREDICTION ACCURACY OVER 17 DAYS, E.G., 15 DECEMBER 2009 TO 31 DECEMBER 2009 FOR MILAN AND 15 DECEMBER 2010 TO 31 DECEMBER 2010 FOR ARUBA

Datasets	Model	Average F1-Score
Aruba [21]	The proposed model	**89.37%**
	DCNN [13]	79.00%
	RNN [17]	85.03%

Given two days, such as d_1 and d_2, we train both of the models using the data sets from day d_1 to day $d_2 - 1$. We use each model to periodically predict activities of day d_2. At time t, we use the binary images $\mathbf{I}^{60}_{t-29}, \ldots, \mathbf{I}^{60}_{t}$ from time $t - 29$ to time t to predict activities from time $t + 1$ to time $t + 10$. We use the sensor data gathered during previous 90 min to predict next activities for upcoming 10 min. Then, at time $t + 10$, we use $\mathbf{I}^{60}_{t-19}, \ldots, \mathbf{I}^{60}_{t+10}$ from time $t - 19$ to time $t + 10$ to predict activities from time $t + 11$ to time $t + 20$. We repeat the routines until the whole activities are predicted.

We prepare Fig. 4 to explain how our model is used to predict next activities from time $t + 1$ to time $t + 10$. First, the model uses the binary images from time $t - 29$ to time t to predict the activity at time $t + 1$, as we have shown in Fig. 4(a). It is noted that the encoded vector $\widehat{\mathbf{e}}_{t+1}$ is estimated. Thus, at time $t + 1$, the model can use $\mathbf{I}^{60}_{t-28}, \ldots, \mathbf{I}^{60}_{t}$ and $\widehat{\mathbf{e}}_{t+1}$ to predict $\hat{y}_{t+2}$, as we have shown in Fig. 4(b). Then, at time $t + 9$, we have a set of encoded vectors, e.g., $\widehat{\mathbf{e}}_{t+1}, \widehat{\mathbf{e}}_{t+2}, \ldots, \widehat{\mathbf{e}}_{t+9}$. Hence, as we have shown in Fig. 4(c), the model can take $\mathbf{I}^{60}_{t-20}, \ldots, \mathbf{I}^{60}_{t}, \widehat{\mathbf{e}}_{t+1}, \ldots, \widehat{\mathbf{e}}_{t+9}$ to $\hat{y}_{t+10}$ at time t + 10. In Table VII, we provide the pseudocodes for predicting next activities from time $t + 1$ to time $t + 10$ when the binary images from time $t - 29$ to time t are provided.

We remind that the nonend-to-end model takes a sequence of one-hot encoded vectors to predict the next activity. At time t, we assume that the activities from time $t - 29$ to time t are given, e.g., $y_{t-29}, \ldots, y_t$. The activity at time $t + 1$ is

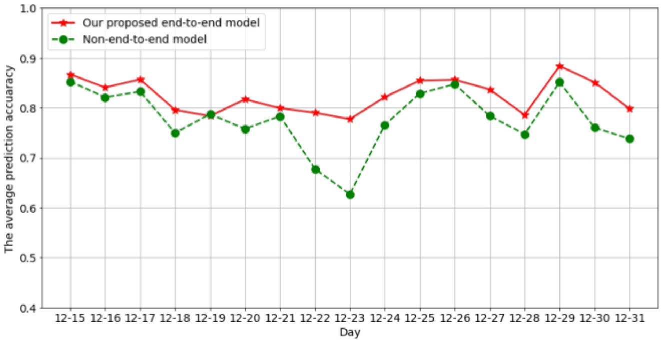


Fig. 5. Average prediction accuracy for Milan [22] in terms of a day.

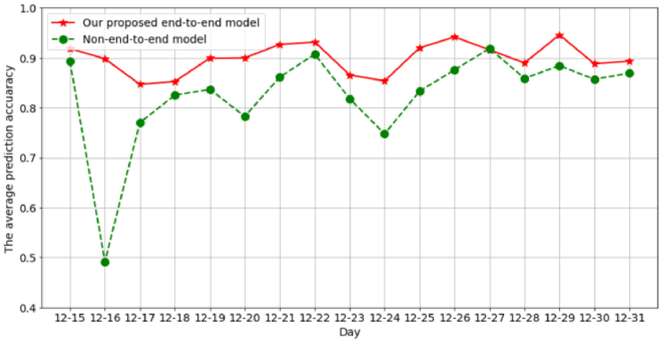


Fig. 6. Average prediction accuracy for Aruba [21] in terms of a day.

TABLE VII
PSEUDO CODE TO PREDICT ACTIVITIES FROM TIME $t+1$ TO TIME $t+10$

The pseudocodes of the prediction algorithm of the proposed model		
Input	$enc(\cdot\,;\theta_{enc})$	the pre-trained encoder
	$reg(\cdot\,;\theta_{reg})$	the pre-trained regressor
	$clr(\cdot\,;\theta_{clr})$	the pre-trained classifier
	$\mathbf{I}_{t-29}^{60},\ldots,\mathbf{I}_{t}^{60}$	a sequence of 30 consecutive binary images from time $t-29$ to time t where each image is generated according to (1).
Output	A sequence of predicted activities $\hat{y}_{t+1},\ldots,\hat{y}_{t+10}$ from time $t+1$ to time $t+10$.	

```
for $i = 0$ to 29
    $\mathbf{e}_{t-i} \triangleq enc(\mathbf{I}_{t-i}^{n};\theta_{enc})$
for $f = 1$ to 10
    $\hat{\mathbf{e}}_{t+f} \triangleq reg(\mathbf{e}_{t-30+f},\mathbf{e}_{t-29+f},\ldots,\mathbf{e}_{t+f-1};\theta_{reg})$
    $\hat{y}_{t+f} = clr(\hat{\mathbf{e}}_{t+f};\theta_{clr})$
    $\mathbf{e}_{t+f} = \hat{\mathbf{e}}_{t+f}$
  return $\hat{y}_{t+1},\ldots,\hat{y}_{t+10}$
```

predicted using one-hot encoded vectors of $y_{t-29},\ldots,y_t$. The activity at time $t+2$ is then predicted using one-hot encoded vectors of $y_{t-28},\ldots,\hat{y}_{t+1}$. These routines are repeated until the nonend-to-end model predicts the activity at time $t+10$.

We set the batch size to ten, and the epoch to five. We use the *Adam* optimizer with a learning rate of 0.001. As an evaluation metric, we use the following prediction accuracy:

$$\text{accuracy} \triangleq \frac{1}{N}\sum_{i}^{N}\mathcal{L}\big(y_i,\hat{y}_i\big) \tag{8}$$

where $\mathcal{L}(x,y)$ is 1 when x is equal to y; otherwise, it is 0, y_i is the known label at time i, $\hat{y}_i$ is the predicted label at time i, and N is the number of samples. We exclude cases in which y_i has no labels when we evaluate the accuracy. Given a day, we run the experiments ten times and evaluated the average prediction accuracy. We consider days from 15 December 2009 to 31 December 2009 for Milan. For Aruba, we consider days from 15 December 2010 to 31 December 2010.

Fig. 5 shows the average prediction accuracy of each model as a function of day using the Milan data set. Then, Fig. 6 shows the average prediction accuracy of each model using the Aruba data set. They show that the proposed model surpasses the nonend-to-end model in terms of the average prediction accuracy.

Table VIII provides the average prediction accuracies. For Milan, the average accuracy of our model was 82.45%, whereas that of the nonend-to-end model was 77.72%. The performance gap between the models was 4.73%. Then, for Aruba, the proposed model and the nonend-to-end model could achieve 89.92% and 82.53%, respectively. The performance gap between them was 7.39%.

We now conduct hypothesis tests to show that our end-to-end model can outperform the nonend-to-end model in terms of the daily prediction accuracy. For this end, we begin to make the following hypothesis:

$$\text{the null hypothesis } \mathrm{H}_0 : u_p = u_n \text{ and}$$
$$\text{the alternative hypothesis } \mathrm{H}_a : u_p > u_n$$

where u_p is the mean of the daily accuracy of our model and u_n is the mean of the daily prediction accuracy of the nonend-to-end model. The daily prediction accuracy for each data set can be obtained from the previous figures. For example, from Fig. 6, we can get the samples during 17 days for Aruba as follows:

$$A_p = \{0.9184, 0.8979, 0.8468, \ldots, 0.9459, 0.8879, 0.8932\}$$
$$A_n = \{0.8926, 0.4918, 0.7701, \ldots, 0.8841, 0.8570, 0.8692\}$$

where A_p is a set of the daily prediction accuracy of our model and A_n is a set of the daily prediction accuracy of the nonend-to-end model. We perform a one-tailed Welch's t-test [51] with a significance level $\alpha = 0.05$. The results are given in Table VIII.

The p-values are 0.00787 and 0.01005 for Aruba and Milan, respectively. They are less than the significance level α. We thus can reject H_0, implying that our end-to-end model achieves the better prediction accuracy rather than the nonend-to-end model does. Then, there is a question that we aim to answer: What factors could make the proposed model surpass the nonend-to-end model?

The answer to this question is that the amount of information used by the proposed model is more abundant than that used by the nonend-to-end model. Namely, the nonend-to-end model does not use the sensor data in the prediction while the proposed model can use them. At time t, we consider the input of each model. The nonend-to-end model takes a sequence of one-hot encoded vectors to predict the next activity at time $t+1$. Hence, this model no longer uses the

TABLE VIII

RESULTS OF THE ONE-TAILED WELCH'S t-TEST. SAMPLE A_p IS A SET OF THE DAILY ACCURACY DURING 17 DAYS WHERE EACH ELEMENT IS OBTAINED USING THE PROPOSED MODEL. SAMPLE A_n IS A SET OF THE DAILY ACCURACY DURING 17 DAYS WHERE EACH ELEMENT IS OBTAINED USING THE NONEND-TO-END MODEL

Datasets	Sample A_p			Sample A_n			Welch's t-test	
	Sample size	Sample mean	Sample variance	Sample size	Sample mean	Sample variance	t	p
Aruba	17	0.8992	0.0299	17	0.8253	0.0950	-2.96663	0.00787
Milan	17	0.8245	0.0327	17	0.7771	0.0596	-2.78659	0.01005

sensor data. In contrast, our model can use the sensor data. As we have shown in Fig. 4, our model can predict the activity at time $t+1$ using $\widehat{\mathbf{e}}_{t+1}$ where $\widehat{\mathbf{e}}_{t+1}$ is estimated from $\mathbf{e}_{t-29}, \ldots, \mathbf{e}_t$. Since each $\mathbf{e}_t$ has the encoded information on the sensor data, $\widehat{\mathbf{e}}_{t+1}$ can also have the encoded information which is used by our model.

We last conduct an experiment to examine the performance of the regressor. We aim to examine the distance between $\widehat{\mathbf{e}}_{t+1}$ and $\mathbf{e}_{t+1}$, where $\widehat{\mathbf{e}}_{t+1}$ depends on the sensor data until time t and $\mathbf{e}_{t+1}$ depends on the sensor data until time $t+1$. If the distance is small, our model can predict the activity at time $t+1$ using the sensor data until time t, which means that our model can use the sensor data to predict the next activities. To measure this distance, we use the relative root mean square error (RRMSE) defined in [43] as follows:

$$\text{RRMSE}(\mathbf{y}, \widehat{\mathbf{y}}) \triangleq \sqrt{\|\mathbf{y} - \widehat{\mathbf{y}}\|_2^2 / \left(n\|\mathbf{y}\|_2^2\right)} \times 100\%$$

where $\mathbf{y}$ is the ground-truth and $\widehat{\mathbf{y}}$ is an output of the regressor, i.e., an estimate of $\mathbf{y}$, $\|\mathbf{y}\|_2$ is the l_2-norm of $\mathbf{y}$ and n is the dimension of $\mathbf{y}$. In [43], a model is considered to be excellent when RRMSE $< 10\%$; good if RRMSE $> 10\%$ and RRMSE $< 20\%$; fair if RRMSE $> 20\%$ and RRMSE $< 30\%$. If RRMSE $> 30\%$, model is poor.

This experiment is conducted as follows. First, we construct all of binary images from the sensor data from 4 November 2010 to 16 December 2010. Second, we use the trained encoder to get corresponding encoded vectors. Third, we train the regressor using the encoded vectors. For date 17 December 2010, we evaluate RRMSE($\mathbf{e}_{t+1}$,$\widehat{\mathbf{e}}_{t+1}$) and RRMSE($\mathbf{e}_{t+2}$,$\widehat{\mathbf{e}}_{t+2}$), respectively.

At time $t+1$, the average RRMSE was 18.27%. Then, the probabilities that the RRMSE $< 5\%$, 10%, and 20% were 27.67%, 47.17%, and 69.33%, respectively. Thus, the performance of the regressor can be good to estimate $\widehat{\mathbf{e}}_{t+1}$ with respect to RRMSE. Second, at time $t+2$, the average RRMSE was 22.87%. Then, the probabilities that the RRMSE $< 5\%$, 10%, and 20.92% were 39.91%, 67.75%, and 69.33%, respectively. The average RRMSE at time $t+2$ is not good as much as that at time $t+1$. This result is obvious because $\widehat{\mathbf{e}}_{t+2}$ was estimated using $\mathbf{e}_{t-28}, \ldots, \mathbf{e}_t$, $\widehat{\mathbf{e}}_{t+1}$. Nevertheless, we can still say that the performance can be fair to estimate $\widehat{\mathbf{e}}_{t+2}$ with respect to RRMSE.

V. CONCLUSION

The HAP models in [19] and [20] predict the next activity by taking a sequence of past activities and their durations. In order to construct this sequence, we recognize the past activities from a sequence of consecutive binary events obtained from ambient sensors. Thus, the prediction by these models is conducted in a nonend-to-end manner. Also, the information to indicate which sensors are activated or not is no longer used in the prediction.

In this article, we proposed an end-to-end model to predict the next activity using the above sequence obtained from ambient sensors. The model consists of three components: 1) *enc*; 2) *reg*; and 3) *clr*, as we have illustrated in Fig. 2. The encoder denoted as *enc* gives an encoded vector that partially preserves the information. The regressor denoted as *reg* predicts the next encoded vector by learning temporal dependencies among consecutive encoded vectors. Finally, the classifier denoted as *clr* takes the predicted encoded vector to predict the next activity.

We used the Aruba [21] and Milan [22] data sets to compare our model with a nonend-to-end model based on LSTM. This nonend-to-end model predicts the next activity by taking the sequence of past activities similar to the models in [19] and [20]. The comparison results shown in Table VI demonstrated that our model outperformed the nonend-to-end model in terms of the average prediction accuracy defined in (8). Specifically, the prediction accuracy of our model was 4.73% higher than that of the nonend-to-end model for Milan. Then for Aruba, our model achieved a better prediction accuracy than the nonend-to-end model by up to 6.69%. These results imply that the information on sensor activation should be used in the prediction to improve the prediction performance.

As future works, we will further develop our framework using an attention module for obtaining better features, leading to the improvement of a prediction performance. We also will extend our framework into cases where data sets were obtained using wearable sensors.

REFERENCES

[1] G. K. Vincent and V. A. Velkoff, *The Next Four Decades: The Older Population in the United States: 2010 to 2050*, U.S. Dept. Commerce, Econ. Stat. Admin., Washington, DC, USA, 2010.

[2] P. Urwyler, R. Stucki, L. Rampa, R. Muri, U. P. Mosimann, and T. Nef, "Cognitive impairment categorized in community-dwelling older adults with and without dementia using in-home sensors that recognize activities of daily living," *Sci. Rep.*, vol. 7, Feb. 2017, Art. no. 42084.

[3] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadaks, and A. Bauer, "Monitoring activities of daily living in smart homes," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 81–94, Mar. 2016.

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

[5] N. Y. Hammerla, S. Halloran, and T. Plotz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2016, pp. 1533–1540.

[6] R. Yao, G. Lin, Q. Shi, and D. C. Ranasinghe, "Efficient dense labelling of human activity sequences from wearables using fully convolutional networks," *Pattern Recognit.*, vol. 78, pp. 252–266, Jun. 2018.

[7] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive Mobile Comput.*, vol. 10, pp. 138–154, Feb. 2014.

[8] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 274–283, Mar. 2010.

[9] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," *Pervasive Mobile Comput.*, vol. 3001, pp. 158–175, Apr. 2004.

[10] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, no. 3, pp. 311–325, Jan. 2010.

[11] 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 *Proc. Towards Integrative Mach. Learn. Knowl. Extraction*, Jul. 2015, pp. 194–205.

[12] D. Singh et al., "Human activity recognition using recurrent neural networks," in *Proc. Int. Cross-Domain Conf. Mach. Learn. Knowl. Extraction*, 2017, pp. 267–274.

[13] M. Gochoo, T.-H. Tan, S.-H. Liu, F.-R. Jean, F. S. Alnajjar, and S.-H. Huang, "Unobtrusive activity recognition of elderly people living alone using anonymous binary sensors and DCNN," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 2, pp. 693–702, Mar. 2019.

[14] T.-H. Tan, M. Gochoo, S.-C. Huang, Y.-H. Liu, S.-H. Liu, and Y.-F. Huang, "Multi-resident activity recognition in a smart home using RGB activity image and DCNN," *IEEE Sensors J.*, vol. 18, no. 23, pp. 9718–9722, Dec. 2018.

[15] J. Medina-Quero, S. Zhang, C. Nugent, and M. Espinilla, "Ensemble classifier of long short-term memory with fuzzy temporal windows on binary sensors for activity recognition," *Expert Syst. Appl.*, vol. 114, pp. 441–453, Dec. 2018.

[16] R. A. Hamad, A. S. Hidalgo, M.-R. Bouguelia, M. E. Estevez, and J. M. Quero, "Efficient activity recognition in smart homes using delayed fuzzy temporal windows on binary sensors," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 2, pp. 387–395, Feb. 2020.

[17] F. A. Machot, S. Ranasinghe, J. Plattner, and N. Jnoub, "Human activity recognition based on real life scenarios," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2018, pp. 3–8.

[18] D. Bouchabou, S. M. Nguyen, C. Lohr, I. Kanellos, and B. Leduc, "Fully convolutional network bootstrapped by word encoding and embedding for activity recognition in smart homes," in *Proc. Int. Workshop Deep Learn. Human Activity Recognit.*, 2021, pp. 111–125.

[19] N. Tax, "Human activity prediction in smart home environments with LSTM neural networks," in *Proc. 14th Int. Conf. Intell. Environ. (IE)*, Jun. 2018,pp. 25–28.

[20] K. Krishna, D. Jain, S. V. Mehta, and S. Choudhary, "An LSTM based system for prediction of human activities with duration," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 147:1–147:31, Jan. 2018.

[21] D. J. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE Intell. Syst.*, vol. 27, no. 1, pp. 32–38, Jan./Feb. 2012. [Online]. Available: http://casas.wsu.edu/ datasets/aruba.zip

[22] D. J. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods Inf. Med.*, vol. 48, no. 5, pp. 480–485, 2009. [Online]. Available: http://casas.wsu.edu/ datasets/ milan.zip

[23] N. Dua, S. N. Singh, and V. B. Semwal, "Multi-input CNN-GRU based human activity recognition using wearable sensors," *Computing*, vol. 103, pp. 1461–1478, Mar. 2021.

[24] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56855–56866, 2020.

[25] H. Cho and S. M. Yoon, "Divide and conquer-based 1D CNN human activity recognition using test data sharpening," *Sensors*, vol. 18, no. 4, p. 1055, Apr. 2018.

[26] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, and J. Zhou, "TSE-CNN: A two-stage end-to-end CNN for human activity recognition," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 1, pp. 292–299, Jan. 2020.

[27] K. D. Garcia et al., "An ensemble of autonomous autoencoders for human activity recognition," *Neurocomputing*, vol. 439, no. 7, pp. 271–280, Jun. 2021.

[28] J. Sena, J. Barreto, C. Caetano, G. Cramer, and W. R. Schwartz, "Human activity recognition based on smartphone and wearable sensors using multiscale DCNN ensemble," *Neurocomputing*, vol. 444, pp. 226–243, Jul. 2021.

[29] X. Gao, H. Luo, Q. Wang, F. Zhao, L. Ye, and Y. Zhang, "A human activity recognition algorithm based on stacking denoising autoencoder and lightgbm," *Sensors*, vol. 19, no. 4, p. 947, Feb. 2019.

[30] M. Zeng et al., "Understanding and improving recurrent networks for human activity recognition by continuous attention," in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, Oct. 2018, pp. 56–63.

[31] V. S. Murahar and T. Plotz, "On attention models for human activity recognition," in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, Oct. 2018, pp. 100–103.

[32] H. Ma, W. Li, X. Zhang, S. Gao, and S. Lu, "AttnSense: Multi-level attention mechanism for multimodal human activity recognition," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3109–3115.

[33] H. Zhang, Z. Xiao, J. Wang, F. Li, and E. Szczerbicki, "A novel IoT-perceptive human activity recognition (HAR) approach using multihead convolutional attention," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1072–1080, Feb. 2020.

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

[35] B. Khaertdinov, E. Ghaleb, and S. Asteriadis, "Deep triplet networks with attention for sensor-based human activity recognition," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2021, pp. 1–10.

[36] A. Wang, G. Chen, C. Shang, M. Zhang, and L. Liu, "Human activity recognition in a smart home environment with stacked denoising autoencoders," in *Proc. 17th Int. Conf. WAIM*, Jun. 2016, pp. 29–40.

[37] A. Wang, S. Zhao, C. Zheng, J. Yang, G. Chen, and C.-Y. Chang, "Activities of daily living recognition with binary environment sensors using deep learning: A comparative study," *IEEE Sensors J.*, vol. 21, no. 4, pp. 5423–5433, Feb. 2021.

[38] R. A. Hamad, M. Kimura, L. Yang, W. L. Woo, and B. Wei, "Dilated casual convolution with multi-head self-attention for sensor human activity recognition," *Neural Comput. Appl.*, vol. 33, pp. 13705–13722, Apr. 2021.

[39] Y. Shen, B. Ni, Z. Li, and N. Zhuang, "Egocentric activity prediction via event modulated attention," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 197–212.

[40] T. Mahmud, M. Billah, M. Hasan, and A. K. Roy-Chowdhury, "Prediction and description of near-future activities in video," *Comput. Vis. Image Understand.*, vol. 210, Sep. 2021, Art. no. 103230.

[41] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1036–1043.

[42] N. Jaouedi, F. J. Perales, J. M. Budaes, N. Boujnah, and M. S. Boublel, "Prediction of human activities based on new structure of skeleton features and deep learning model," *Sensors*, vol. 20, no. 17, p. 4944, 2020.

[43] M.-F. Li, X.-P. Tang, W. Wu, and H.-B. Liu, "General models for estimating daily global solar radiation for different solar radiation zones in mainland China," *Energy Convers. Manage.*, vol. 70, pp. 139–148, Apr. 2013.

[44] S. M. Lee and B. Edmonston, "Living alone among older adults in Canada and the U.S.," *Healthcare*, vol. 7, no. 2, p. 68, 2019.

[45] S. Toot, T. Swinson, M. Devine, D. Challis, and M. Orrell, "Causes of nursing home placement for older people with dementia: A systematic review and meta-analysis," *Int. Psychogeriatr.*, vol. 29, no. 2, pp. 195–208, Feb. 2017.

[46] Z. A. Khan and W. Sohn, "Abnormal human activity recognition system based on R-transform and kernel discriminant technique for elderly home care," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1843–1850, Nov. 2011.

[47] G. Sebestyen, I. Stoica, and A. Hangan, "Human activity recognition and monitoring for elderly people," in *Proc. IEEE 12th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2016, pp. 341–347.

[48] M. A. A. Al-Qaness, A. Dahou, M. A. Elaziz, and A. M. Helmi, "Multi-ResAtt: Multilevel residual network with attention for human activity recognition using wearable sensors," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 144–152, Jan. 2023.

[49] A. M. Helmi, M. A. A. Al-Qaness, A. Dahou, R. Damasevicius, T. Krilavicius, and M. A. Elaziz, "A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human activity recognition using smartphone sensors," *Entropy*, vol. 23, no. 8, p. 1065, Aug. 2021.

[50] N. Rashid, B. U. Demirel, and M. A. A. Faruque, "AHAR: Adaptive CNN for energy-efficient human activity recognition in low-power edge devices," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13041–13051, Aug. 2022.

[51] G. D. Ruxton, "The unequal variance t-test is an underused alternative to student's t-test and Mann-Whitney U test," *Behav. Echol.*, vol. 17, no. 4, pp. 688–690, 2006.

Sangjun Park received the B.S. degree in computer engineering from Chungnam National University, Daejeon, South Korea, in 2009, and the Ph.D. degree in electrical engineering and computer science from Gwangju Institute of Science and Technology, Gwangju, South Korea, in 2019.

Since 2020, he has been working with the Electronics and Telecommunications Research Institute, Daejeon. His research interests include information theory, numerical optimization, compressed sensing, blockchain, deep-neural networks, and finite-state machine.

Hyung Ok Lee received the B.S., M.S., and Ph.D. degrees in computer engineering from Chonnam National University, Gwangju, South Korea, in 2006, 2008, and 2015, respectively.

Since 2016, he has been a Senior Member of Engineering Staff with the Electronics and Telecommunications Research Institute, Daejeon, South Korea. His current research interests include microservice, energy IoT, and big data.

Yu Min Hwang (Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Wireless Communications Engineering, Kwangwoon University, Seoul, South Korea, in 2012 and 2018, respectively.

He was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, Western University, London, ON, Canada, from 2019 to 2020. He is currently with the Electronics and Telecommunications Research Institute, Daejeon, South Korea. His research interests include Internet of Energy, deep learning, activity recognition, and anomaly detection.

Seok-Kap Ko received the B.S., M.S., and Ph.D. degrees in information telecommunication engineering from Soongsil University, Seoul, South Korea, in 1997, 2002, and 2009, respectively.

Since 2008, he has been a Principal Research Engineer with the Electronics and Telecommunications Research Institute, Daejeon, South Korea. His research interests include machine learning for energy management systems.

Byung-Tak Lee received the B.S. degree from Yonsei University, Seoul, South Korea, in 1992, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1994 and 2000, respectively.

From 2000 to 2004, he was a Principal Research Engineer with LG Electronics, Seoul, South Korea, where he was engaged in Tbps optical transmission systems. Since 2005, he has been a Principal Research Engineer with the Electronics and Telecommunications Research Institute, Daejeon. His current research interests include the Internet of Things, deep anomaly detection, and system optimization.