2024 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT) | 979-8-3503-6944-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/DCOSS-IoT61029.2024.00063

Human Activity Recognition Using Spectrograms of Binary Motion Sensor Data

Nima Seyedtalebi
Department of Computer Science
University of Kentucky
Lexington, USA
nima.seyedtaleb@uky.edu

Simone Silvestri
Department of Computer Science
University of Kentucky
Lexington, USA
silvestri@cs.uky.edu

***Abstract*—Human activity recognition is at the basis of several applications in the smart living domain, such as energy management, elder care, and health management. Human activity recognition research can be divided into two categories, depending on the type of sensors used: wearable sensors, such as those found in mobile phones and smart watches, and ambient sensors, such as motion sensors or cameras placed in the environment. Among ambient sensors, *binary sensors* are often perceived as less invasive than sensors that collect video, audio, or biometric data. However, the performance of classifiers trained on binary sensor data is often lower since the data inherently contains less information. In this paper, we propose a non-intrusive human activity recognition framework that only exploits binary sensor data and results in high classification accuracy. Our approach is inspired by audio and image processing applied to binary sensors. Specifically, we exploit the Short-Time Fourier Transform (STFT) to extract features from binary data. These features are used to train a hybrid machine learning model which pairs Convolutional Neural Network (CNN) with a Long-Short-Term Memory (LTSM) architecture. We use a real dataset of human activities monitored through binary sensor data for evaluating the impact of the features on classifier performance. Results show that the proposed method significantly outperforms state-of-the-art solutions, requiring minimal training data needed to achieve a given level of accuracy.**

***Index Terms*—Internet of Things, Smart Home, Human Activity Recognition, Spectrogram, Short-Time Fourier Transform**

I. Introduction

Human activity recognition (HAR) has been an area of active research, with applications including elder care and health management [10, 3, 11], energy use predictions [1, 5, 21, 15, 7], smart home/smart environment [4], security and surveillance, indoor navigation, retail, and others [13]. Generally, HAR approaches use data from sensors, such as wearable sensors (e.g., accelerometers, GPS transceivers, smart phones) and ambient sensors (e.g., motion sensors, temperature sensors, switches, cameras, and microphones) [6]. Through these sensors, the user interaction with the environment is observed. Wearable sensors can provide information on the user's movements and location, while ambient sensors can detect opened doors and drawers, noises made in the environment, and the user motion through cameras and infrared sensors. This data is used to infer the activity that the user is performing. Examples of these activities include sleeping eating, working, washing disches, etc.

An alternative approach to wearable sensors is the use of ambient sensors with binary output. Such sensors are less invasive and do not reveal privacy sensitive information such as cameras. The Center for Advanced Studies in Adaptive Systems (CASAS) at Washington State University [8] produced a "smart home in a box" system that includes infrared motion sensors, temperature sensors, light level sensors, and a small server for processing and storing data. The CASAS produced over 30 datasets, many of which are publicly available at their website.[1] These works were undertaken in the late 2000s and early 2010s before the rise of deep learning and use what are now called traditional machine learning techniques. They required a domain expert to create features, a more compact representation of the raw data that captures significant properties. The rise of deep learning encouraged new work with the existing datasets as researchers started applying deep neural networks to learn which features to use for classification instead of specifying them beforehand [16]. As of 2023, deep-learning models predominate.

Classification algorithms usually require the input data to be transformed into *features*, a condensed representation that captures the most statistically significant components of the input. This transformation step can be further broken down into segmentation and encoding. During *segmentation*, the input sequence is partitioned so that each resulting subsequence represents a single example to be associated with a label. The subsequences are used to compute features during *encoding*. When combined with labels, the examples can be used as input to a statistical model that predicts the most likely label for each example. Different segmentation and encoding schemes can significantly affect the final classification accuracy [12].

A. Related Works

Several efforts on HAR research have focused on the use of mobile/wearable sensors. Accelerometer and gyroscope data were used in [14], and [9] uses phase shift data from RFID tags. The use of Fourier-related transforms for compression and feature selection has also been adopted. For example,

[1] https://casas.wsu.edu/

2325-2944/24/$31.00 ©2024 IEEE
DOI 10.1109/DCOSS-IoT61029.2024.00063

the discrete cosine transform (DCT) is central to the JPEG compression scheme [24]. These approaches adopt mobile and wearable sensors which are inherently invasive and thus may make users reluctant to use such devices. The accuracy of classification systems that use wearable sensors with continuous output is often higher because of the richer training data.

Works in HAR with binary ambient sensors can be grouped by how they segment and encode the raw data and classifier architecture as in table I.

Ref	**Datasets**	**Segmentation**	**Encoding**	**Best Arch.**
[18]	Aruba	Event, Dynamic	DWN	CNN2D
[27]	Aruba	Event, Dynamic	DWN	CNN-LSTM
[25]	De novo	Time, Window	Activations	CNN1D
[11]	De novo	Time, Window	FTW	CNN1D-LSTM
[17]	CMK3	Time	Activations	LSTM
[10]	Aruba	Time, Window	Activations	CNN
[22]	Kasteren	Time, Window	Activations	LSTM

Table I
RELATED WORK IN AMBIENT SENSOR-BASED HAR. IN THE "DATASETS" COLUMN, "CMK3" REFERS TO THE CASAS CAIRO, MILAN, AND KYOTO3 DATASETS

Segmenting by time using a fixed-size sliding window appears to be the most common method [25, 11, 19, 22]. An event-based segmentation scheme was used in [20, 18, 27] wherein sensor events are used to define segment boundaries. Time-based segmentation schemes usually require resampling the input data at a uniform rate. Correlation-based approaches were used in [27] and [18] to dynamically determine segment boundaries instead of a fixed-size window.

Classifier architectures often constrain the shape of the input or vice versa. For example, the 2D CNNs often used for image processing take 3D input: one 2D array of intensity values for each color channel. In [25], [26], and [12], different encoding schemes were examined and it was found that the choice of encoding can have a significant impact on accuracy and resource requirements.

The scheme listed as "Activations" in table I refers to using $n \times m$ arrays for binary sensor data where n is the number of time windows and m is the number of sensors. Each entry in the activation matrix A_{nm} represents the number of times sensor m was activated during the n-th time window. The other encoding schemes in Table I were created to address some challenges associated with using raw activations. The authors of [18] mention two such challenges:

- Activity segments of different lengths can be problematic for CNNs because the data must be zero-padded to a uniform length and the CNN cannot ignore the padding, perhaps (mis)interpreting it as useful information;
- Extraction of features that can capture behavioral semantics and spatio-temporal correspondences at the same time.

Directed Weighted Networks, or DWNs, use a method called *stigmergy* inspired by ant colonies to encode movement between different sensor locations. Stigmergy is an emergent modeling paradigm where independent agents coordinate indirectly by traces left in the environment [18, 27]. Ant colonies are a canonical example of stigmergy in action. Each ant deposits marker chemicals called *pheromones* that attract other ants. The ants are sensitive to the concentration of pheromones which are volatile and thus, their concentration in the environment decreases over time. When presented with multiple tracks of pheromones, ants will preferrentially follow the one with the highest concentration. Thus, over time the colony is able to act in a coordinated manner without direct communication. DWNs encode a digital version of these stigmergic tracks by using sensor activations over time to determine where "pheremones" are concentrated.

B. Contributions

Despite the existing efforts in HAR using binary senors, they generally suffer from relatively low accuracy. In this work, we propose a novel method for feature extraction from binary sensors based on Short-Time Fourier Transform (STFT). These features are used to train a hybrid machine learning model which pairs Convolutional Neural Network (CNN) with a Long-Short-Term Memory (LTSM) architecture. We perform an extensive evaluation using the "Aruba" dataset produced by the Center for Advanced Studies in Adaptive Systems (CASAS). Results show that the proposed method significantly outperforms state-of-the-art solutions by achieving a significantly higher classification accuracy of the considered activities.

II. PROBLEM FORMULATION AND SYSTEM MODEL

We consider a set of H ambient sensors deployed in a smart home. These sensors have a binary output and are loosely synchronized. Formally, we refer to $s_i[t]$ as the *state* of sensor i at time t. $s_i[t]$ is a discrete-time vector-valued function. Let $F_i[\tau]$ be a feature vector that is derived from one or more state vectors for sensor i at time τ. We discuss in the next section our approach to extract these features. We consider a set of label L, containing the user activities we plan to classify. Our problem is the following:

> Given a sequence of state vectors for each sensor and the set of activity labels L, predict the most likely label $l \in L$ for each $F(\tau)$.

The prediction is performed based on a set of training examples, where each example is a pair of the form $(F[\tau], l)$.

III. PROPOSED SOLUTION

The approaches to segmentation in the literature can be classified by whether they define segment boundaries using time intervals or sensor state changes. The two approaches are linked because the time of state changes is known, so each pair of state changes could be used to define a time interval. If segments are defined in terms of time intervals other than those such as a fixed-length sliding window, then resampling of the discrete-valued $s_i[t]$ will be necessary. In either case, the segments can be defined mathematically using the concept from signal processing called *window functions*. These functions are defined so they have a value of zero everywhere except for an interval where they take on values in the range $[0, 1]$. Multiplying a sequence by a window function has the effect of selecting a subset of the sequence. If $w[t-\tau]$ is a time-shifted window function and $s_i[t]$ represents the state of sensor i at time t, then segmentation can be represented by:

$$s_i[t]w[t-\tau] \tag{1}$$

The encoding step takes the segmented input data and outputs features. This can be modeled as the application of a function E to the segmented input data to produce feature F_i:

$$F_i[\tau] = E[s_i[t]w[t-\tau]] \tag{2}$$

Our solution differs substantially from the others presented in Table I in how the features are encoded. With the exception of FTWs, the feature encodings in Table I produce one or two-dimensional feature vectors. Our approach treats each sensor as a separate channel independent of the others, much as color images can be processed as three independent color channels. For each sensor, we segment the data using an overlapping fixed-length window. Then, each segment is encoded by applying the discrete Fourier transform. The combination of overlapping window and Fourier transform is called the *Short-Time Fourier Transform* and is often used to study signals with time-varying frequency spectra such as audio. The magnitude-squared output of the STFT is called a *spectrogram*.

The STFT can be defined as the discrete-time Fourier transform (DTFT) of a sampled signal $x[t]$ multiplied by a shifted window function. We consider a symmetric window function on the range $[-T/2, T/2]$. The size of the range, T, is often called the length of the window. The formulation presented here is adopted from [23]. Let D be a constant that determines how many samples the window moves between DFT calculations. If the sampled signal $x[t]$ has a length of n, the discrete STFT is given by:

$$S[\tau, \omega] = \sum_{t=0}^{n-1} x[t]\overline{w[t-\tau D]}\, e^{-i\omega t/n} \tag{3}$$

where $\overline{w[t-\tau D]}$ is the complex conjuate of the window function. The STFT combines the segmenting and encoding steps into a single operation. If $s_i[t]$ is each sensor's state as a function of time, the features for each sensor as a function of window index τ are obtained by applying the STFT:

$$F_i[\tau] = \sum_{t=0}^{n-1} s_i[t]\overline{w[t-\tau D]}\, e^{-i\omega t/n} \tag{4}$$

The classifier portion of the proposed system is a hybrid CNN-LSTM architecture. We have experimentally verified that such an architecture yields better results compared to a plain CNN or LSTM architecture. The architecture is shown in Figure III. The block labels in the block diagram correspond to the function signature for each layer. Dimensions for arrays at each step are given for spectrogram and DWN feature encodings (DWN is used for comparison, as explained in the following section). n refers to the number of examples of training data, and $ws//2$ is the integral part of *window size* divided by 2. The dimensions are the same after the Conv1D

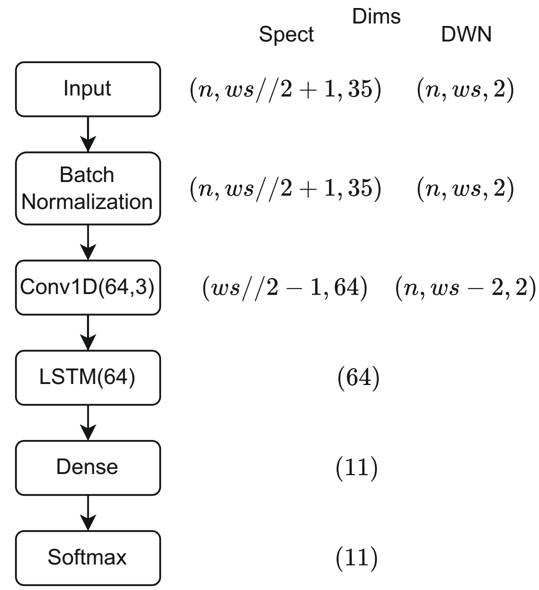


Figure 1. Hybrid CNN-LSTM Classifier architecture.

for all models tested, and the size of the output is the number of labels.

IV. Results

In this section we provide an extensive evaluation of the proposed method versus state of the art solutions. In the following, we first describe the dataset, then the experimental setup, and finally present the results.

A. Dataset

We tested the performance of the proposed spectrogram feature encoding using a public dataset called "Aruba," one of the datasets produced by the CASAS Smart Home-in-a-Box project [8]. The Smart Home-in-a-Box system used wireless infrared motion sensors, contact sensors on doors and drawers, temperature sensors, and light-level sensors placed throughout the home. The sensors communicated with a central server by sending messages that included a timestamp, sensor id, message type, and payload. For example, the infrared motion sensors would send messages saying "ON" when motion is first detected and "OFF" after not detecting any motion for 1.25 seconds.

The Aruba dataset was collected in the home of an adult volunteer who lived alone but had visitors on a regular basis. The timestamps range from 2010-11-04T00:03:50 to 2011-06-11T23:20:35, a span of 219 days. We resampled the raw data to a uniform rate of one sample per second for each of the binary sensors in the Aruba smart home, yielding an array with dimensions (19005405,35). Figure 2 shows where the sensors were placed in the smart home where the Aruba dataset was collected. The dataset is distributed as a text file with one event record per line. Activities are annotated by noting when they

Figure 2. Location of motion sensors in smart home. The dotted ellipses with text in the center represent infrared motion sensors with a wide area of detection. The other sensors, represented by small circles with text above or below, are sensitive to motion in a one meter circle directly below the sensor.

start and end. The annotations and their relative frequency in the dataset are:

- No annotation (55.2 %)
- Relax (21.3 %)
- Meal_Preparation (17.9 %)
- Sleeping (2.04 %)
- Work (1.02 %)
- Eating (0.959 %)
- Housekeeping (0.656 %)
- Wash_Dishes (0.647 %)
- Enter_Home (0.126 %)
- Leave_Home (0.121 %)
- Bed_to_Toilet (0.083 %)

The dataset is typical for HAR data in that it is highly unbalanced, even if the un-annotated portion is discarded. We tried our experiments with and without this unannotated data included. Where we used the un-annotated data, we assigned all un-annotated events a label of "Other".

B. Experimental Setup

We used the Aruba dataset to test the proposed activity recognition system versus several other state-ot-the-art encodings from the literature. The various implementations used a sliding window for segmentation and the same overall network architecture for fair comparisons. Classification accuracy was the metric chosen to evaluate performance. We used a leave-one-day-out cross-validation scheme for all experiments where each day's worth of data is held out as a test set in turn. We randomly sampled which days of the dataset to include in the experiments. All experiments were run on a consumer-grade desktop computer with an 8-core CPU, 128 GB of RAM, and an NVidia RTX 3080 Ti GPU. The preprocessing steps and experiments were implemented using Tensorflow and other Python-based tools inside of a container to improve repeatability.

In all experiments, our approach is referred to as "Spect", given its spectrogram nature. We consider the following comparison approaches: *DWN*, *Activations*, and *Last*, using the same time-based segmentation method and classifier architecture as the spectrogram-based models. The DWN encoding was used in [27, 18] and can be considered an extension of the Activations method used in [22, 17, 25, 10]. DWN extends the Activations method by adding an exponential decay to the summation. Returning to the ant colony analogy, if one unit of pheromone is deposited for each unit of time a sensor is active and ρ percent of the pheromone evaporates each time step, then the concentration or intensity of pheromone remaining at time t is:

$$I = \sum_{t_s}^{t_e-1} (1-\rho)^{t_e-t-1} \tag{5}$$

t_s and t_e are the start and end times of the activity segment, respectively.

C. Experiments Performed

We performed several experiments varying different model hyperparameters to evaluate the proposed system and compare it with the current state of the art. We tested varying numbers of days of training data, window size, and inclusion/exclusion of "Other" labels for models using the "spectrogram," "DWN," "Activations," and "Last" feature encodings. Additionally, we tried varying hyperparameters specific to the proposed spectrogram feature encoding such as the amount of overlap in the sliding window and the shape of the window function.

D. Impact of Training Data Size

We first investigate the accuracy versus the amount of training data. The objective of this experiment is to show how the classification performance change with respect to the amount of data available for training. Figure 3 shows the average accuracy versus the number of days of training data with the "Other" labels included. All approaches stabilize in accuracy with 5-10 days of data. With the "Other" label included, the comparison approaches converge to an accuracy around 75%. Conversely, our spectrogram-based model is able to achieve an accuracy above 90%. This is due to our solution producing higher-dimensional features that are effectively compressed by the application of a Fourier-related transform. Our approach significantly out performs the comparison solutions providing 15% higher accuracy and a comparable number of training days to converge. When the "Other" labels are not included, the gap in accuracy between the spectrogram method and the others decreases to about 5 %, and the spectrogram models converge to their maximal accuracy more quickly.

E. Impact of Window Size

In the next experiment we focus on the impact of the window size on the classification accuracy. We use fifteen days worth of data for training and vary the length of the window used for segmentation, for all approaches. Our rationale is that for all the methods tested, fifteen days is large enough

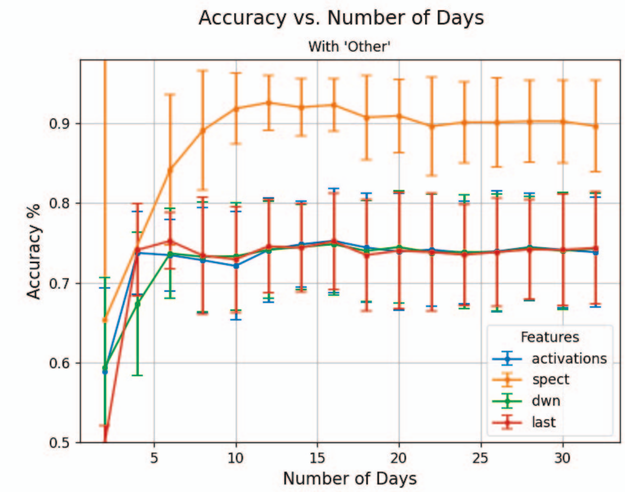


Figure 3. Number of days of training data versus accuracy. Error bars show an interval $[-\sigma, \sigma]$

to stabilize the accuracy, and the accuracy is close to the maximum seen for each of the different methods. We vary the window size. This setting dictates the maximum resolution in frequency as described in [2] and others. This can be appreciated intuitively by the inverse relationship between period and frequency.

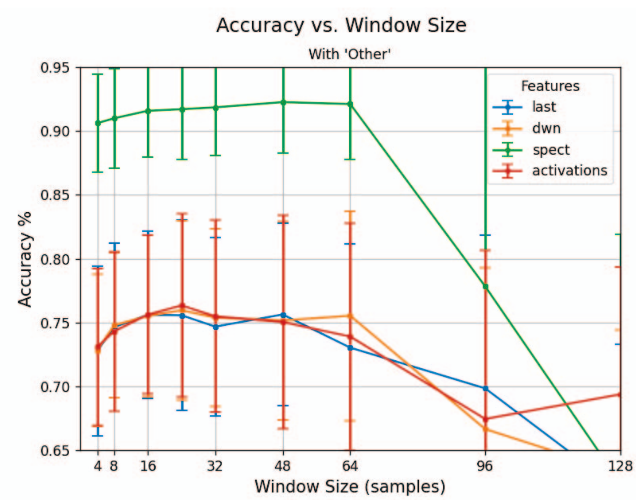


Figure 4. Window size vs accuracy.Number of days (of data) versus accuracy. Error bars show an interval $[-\sigma, \sigma]$

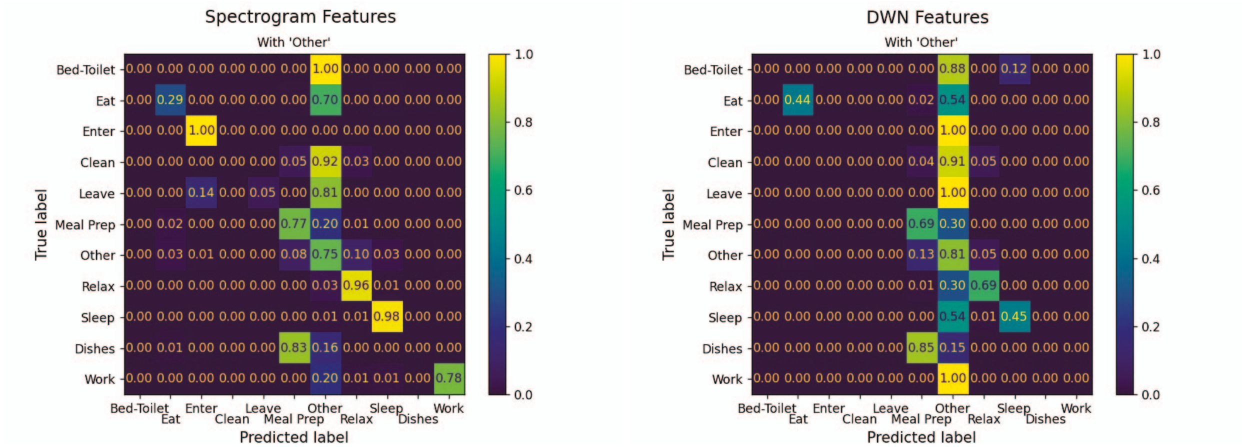
Figure 4 shows the results when the "Other" label is included. Our approach provides a higher accuracy in these experiments as well. All approaches suffer when the window size increases beyond 64 samples. At a sample rate of 1 Hz, 64 samples corresponds to 64 seconds of sensor data. Increasing the window length increases frequency resolution, especially for lower frequencies. However, with the improved frequency resolution comes a corresponding loss of temporal resolution. With a longer window, the labels in the training data are applied to longer periods of time. As an example, if activities A and B are interleaved in time and activity B only occurs for a few seconds within much longer blocks of activity A, insufficient temporal resolution would make two closely-spaced occurrences of B indistinguishable from one longer occurrence. Overall, these results show that the setting of the time window is not critical, and any value less than 64 would provide high accuracy. Our approach outperforms the state-of-the-art with 15% higher accuracy. As with the experiments varying data size, when the "Other" labels are excluded, the gap in performance is reduced to around 5 %.

Note that the amount of overlap between windows has a also impact on the performance of our spectrogram model. However, more overlap resulted in a modest effect on the amount of training data required. The final run of experiments to produce Figures 3 and 4 used a window with 50 percent overlap for all of the spectrogram-based models. Increasing the overlap beyond 50 percent yielded no benefit to accuracy but increased training time due to the larger number of examples. We also considered changing the shape of the window function. This also has minimal impact on the accuracy. A simple rectangular window, sometimes called a "boxcar" window, produced consistently good performance. Triangular, cosine, and exponential windows performed worse than rectangular windows, but they were all within a few percent of each other.

F. Effect of Unannotated Data

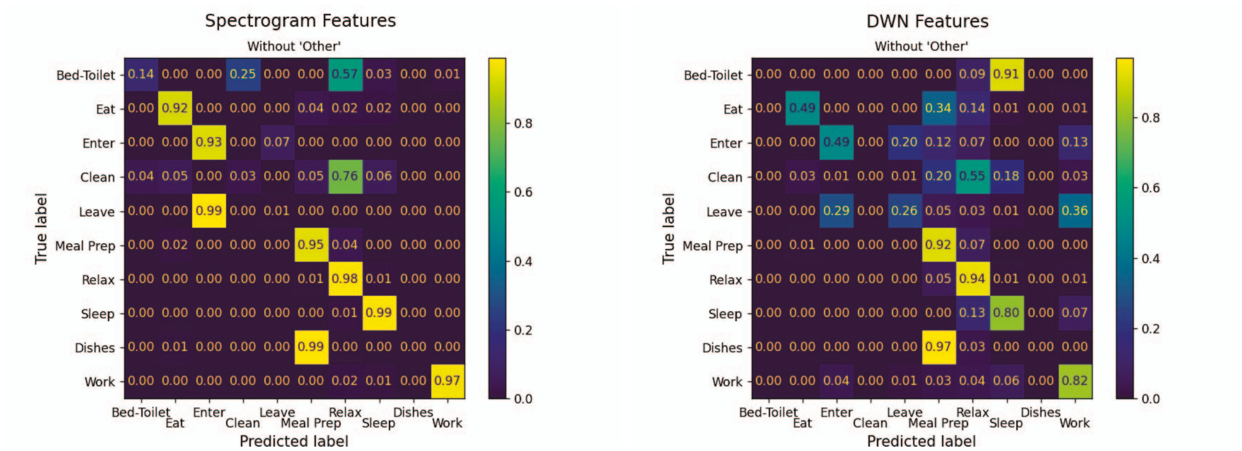
The Aruba dataset has gaps in the annotations; some events are recorded but not annotated with an activity label. We investigated the effect of including these unannotated events by assigning them all the label "Other." The presence of the Other category makes the classification problem more difficult, because there may be significant variation in what the examples in this category look like. Additionally, the relative frequency of the Other class is very high, making the dataset more unbalanced. To explore the impact of the Other category, we generated confusion matrices from the predictions of our spectrogram models and DWN model. We also show the confusion matrices without the Other category, for comparison. The confusion matrices for the models that include the "Other" label are shown in Figures 5 (a) and (b).

Both of the confusion matrix of Spec and DWN exhibit a skew toward the Other class as expected. The column associated with Other contains relatively high values, which means that the classifier often mislabeled activities as Other. As expected this effect is more pronounced where the non-other class has a very low relative frequency. This is evident in Figures 5 (a) and (b). Both models label incorrectly most occurrences of "Bed-to-toilet", "Enter", "Dishes", "House-keeping". Nevertheless, this result show the superiority of our approach with respect to DWN. The spectrogram model has fewer rows in the "Other" column, and better distributes the labels across the diagonal (correct labeling). This suggests that our approach is less affected by highly unbalanced datasets. This results also show that both models have trouble distin-



(a) (b)

Figure 5. Confusion matrices of Spect (a) and DWN (b) when the "Other" class is included.



(a) (b)

Figure 6. Confusion matrices of Spect (a) and DWN (b) when the "Other" class not included.

guishing between activities that typically occur in the same part of the smart home. For example, both models struggled to correctly distinguish between "Meal Prep" and "Dishes".

We now show the confusion matrices when the data labelled Other is not included. Figures 6 (a) and (b) show the results. Both approaches improve the classification. However, the DWN model shows a tendency toward diffuse mislabeling. Conversely, our spectrogram model is able to classify more activities correctly, resulting in a higher overall accuracy.

V. Conclusion and Future Work

In this paper, we describe and test a novel human activity recognition framework that uses spectrograms of binary sensor data and a CNN-LSTM classifier to predict the activities taking place inside a smart home. This is the first time that spectral features have been used with binary sensors. Our experiments showed that the spectrogram-based models outperformed the state-of-the-art in all of our experiments. Our results also show that all approaches, including ours, suffer labeling activities with a small number of samples, as well as a decrease in accuracy when a large unlabeled class is present. These remain open problems that can be addressed in future works.

VI. Acknowledgements

This paper was partially supported by the NSF CAREER grant Nr. 1943035.

References

[1] Musaed Alhussein, Khursheed Aurangzeb, and Syed Irtaza Haider. "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting". In: *IEEE Access* 8 (2020), pp. 180544–180557.

[2] J. Allen. "Short term spectral analysis, synthesis, and modification by discrete Fourier transform". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25.3 (June 1977), pp. 235–238.

[3] Damien Bouchabou et al. “A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning”. In: *Sensors* 21.18 (Sept. 2021), p. 6037.

[4] Enrico Casella et al. “Hierarchical syntactic models for human activity recognition through mobility traces”. In: *Personal and Ubiquitous Computing* 24 (2020), pp. 451–464.

[5] Enrico Casella et al. “Hvac power conservation through reverse auctions and machine learning”. In: *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2022, pp. 89–100.

[6] Liming Chen and Chris D. Nugent. “Sensor-Based Activity Recognition Review”. In: *Human Activity Recognition and Behaviour Analysis: For Cyber-Physical Systems in Smart Environments*. 2019, pp. 23–47.

[7] Jackson Codispoti et al. “Learning from non-experts: an interactive and adaptive learning approach for appliance recognition in smart homes”. In: *ACM Transactions on Cyber-Physical Systems (TCPS)* 6.2 (2022), pp. 1–22.

[8] Diane J. Cook et al. “CASAS: A Smart Home in a Box”. In: *Computer (Long Beach Calif)* 46.7 (July 2013), pp. 10.1109/MC.2012.328.

[9] Xiaoyi Fan, Wei Gong, and Jiangchuan Liu. “TagFree Activity Identification with RFIDs”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2.1 (Mar. 2018), 7:1–7:23.

[10] Munkhjargal Gochoo et al. “Unobtrusive Activity Recognition of Elderly People Living Alone Using Anonymous Binary Sensors and DCNN”. In: *IEEE J. Biomed. Health Inform.* (2018), pp. 1–1.

[11] Rebeen Ali Hamad et al. “Efficient Activity Recognition in Smart Homes Using Delayed Fuzzy Temporal Windows on Binary Sensors”. In: *IEEE Journal of Biomedical and Health Informatics* 24.2 (Feb. 2020), pp. 387–395.

[12] Harish Haresamudram, David V. Anderson, and Thomas Plötz. “On the role of features in human activity recognition”. In: *Proceedings of the 2019 ACM International Symposium on Wearable Computers*. Sept. 2019, pp. 78–88.

[13] Zawar Hussain, Michael Sheng, and Wei Emma Zhang. “Different Approaches for Human Activity Recognition: A Survey”. In: *Journal of Network and Computer Applications* 167 (Oct. 2020), p. 102738.

[14] Wenchao Jiang and Zhaozheng Yin. “Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks”. In: *Proceedings of the 23rd ACM international conference on Multimedia*. Oct. 2015, pp. 1307–1310.

[15] Atieh R Khamesi, Eura Shin, and Simone Silvestri. “Machine learning in the wild: The case of user-centered learning in cyber physical systems”. In: *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE. 2020, pp. 275–281.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (May 2015), pp. 436–444.

[17] Daniele Liciotti et al. “A sequential deep learning application for recognising human activities in smart homes”. In: *Neurocomputing* 396 (July 2020), pp. 501–513.

[18] Houda Najeh, Christophe Lohr, and Benoit Leduc. “Convolutional Neural Network Bootstrapped by Dynamic Segmentation and Stigmergy-Based Encoding for Real-Time Human Activity Recognition in Smart Homes”. In: *Sensors* 23.4 (Feb. 2023), p. 1969.

[19] Francisco Javier Ordóñez and Daniel Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition”. In: *Sensors (Basel)* 16.1 (Jan. 2016), p. 115.

[20] Worrakit Sanpote et al. “Deep Learning Approaches for Recognizing Daily Human Activities Using Smart Home Sensors”. In: Mar. 2023, pp. 469–473.

[21] Eura Shin et al. “A user-centered active learning approach for appliance recognition”. In: *2020 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE. 2020, pp. 208–213.

[22] Deepika Singh et al. “Convolutional and Recurrent Neural Networks for Activity Recognition in Smart Environment”. In: 2017, pp. 194–205.

[23] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature methods* 17.3 (2020), pp. 261–272.

[24] Gregory K Wallace. “THE JPEG STILL PICTURE COMPRESSION STANDARD”. In: (1991).

[25] Aiguo Wang et al. “Activities of Daily Living Recognition With Binary Environment Sensors Using Deep Learning: A Comparative Study”. In: *IEEE Sensors Journal* 21.4 (Feb. 2021), pp. 5423–5433.

[26] Aiguo Wang et al. “Comparison of Feature Extraction Techniques for Ambient Sensor-based In-home Activity Recognition”. In: *2022 International Conference on Networking and Network Applications (NaNA)*. Dec. 2022, pp. 1–6.

[27] Zimin Xu, Guoli Wang, and Xuemei Guo. “Online Activity Recognition Combining Dynamic Segmentation and Emergent Modeling”. In: *Sensors* 22.6 (Mar. 2022), p. 2250.