# A Window-Based Sequence-to-One Approach with Dynamic Voting for Nurse Care Activity Recognition Using Acceleration-Based Wearable Sensor

Yiwen Dong
ywdong@stanford.edu
Stanford University
Stanford, California

Jingxiao Liu
Stanford University
Stanford, California

Yitao Gao
Stanford University
Stanford, California

Sulagna Sarkar
Carnegie Mellon University
Pittsburgh, Pennsylvania

Zhizhang Hu
University of California, Merced
Merced, California

Jonathon Fagert
Carnegie Mellon University
Pittsburgh, Pennsylvania

Shijia Pan
University of California, Merced
Merced, California

Pei Zhang
Carnegie Mellon University
Pittsburgh, Pennsylvania

Hae Young Noh
Stanford University
Stanford, California

Mostafa Mirshekari
Stanford University
Stanford, California

## ABSTRACT

This paper introduces a window-based sequence-to-one approach with dynamic voting for nurse care activity recognition using acceleration-based wearable sensors. Nurse care activity recognition is an essential part of ensuring high quality patient care and providing constructive and concrete feedback to the care team. Some of the current sensing approaches for activity recognition include vision-based sensing and non-wearable RF sensing. However, their application is limited in real-life scenarios due to restrictive factors such as perceived privacy and sensitivity to specific occupant paths. To overcome these limitations, acceleration-based wearable sensing have been introduced in recent works. However, the duration distribution of nursing activity instances are biased and skewed. This skewness leads to imbalanced datasets which will result in low performance for the common predictive models. Further, uncertainties such as ambient noise and environmental factors affect the signals and thus can potentially reduce the activity recognition performance. To overcome the first challenge, we separate the signals into short windows with adaptive overlapping ratios for activity instances having different lengths, which balances the label distribution due to event length variations. Further, we use a multi-layer Long Short-Term Memory (LSTM) model to predict nursing activities of each sliding window and introduce a voting-based scheme for complementing the predictions across the signal windows and addressing the uncertainty challenge. We validate our approach through participation in "The 2nd Nurse Care Activity Recognition Challenge Using Lab and Field Data" as team HealthyVibes. On the challenge dataset our model achieves 97.4% and 43.9% accuracy for training and validation, respectively.


*UbiComp/ISWC '20 Adjunct, September 12–16, 2020, Virtual Event, Mexico*

ACM ISBN 978-1-4503-8076-8/20/09…$15.00
https://doi.org/10.1145/3410530.3414336

## CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Applied computing → Health care information systems**.

## KEYWORDS

Wearable Sensors; Activity Recognition; Sequence Model; Nurse Care

**ACM Reference Format:**
Yiwen Dong, Jingxiao Liu, Yitao Gao, Sulagna Sarkar, Zhizhang Hu, Jonathon Fagert, Shijia Pan, Pei Zhang, Hae Young Noh, and Mostafa Mirshekari. 2020. A Window-Based Sequence-to-One Approach with Dynamic Voting for Nurse Care Activity Recognition Using Acceleration-Based Wearable Sensor. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC '20 Adjunct), September 12–16, 2020, Virtual Event, Mexico.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3410530.3414336

## 1 INTRODUCTION

Nursing care is an essential component of medical care in a variety of settings - home care, assisted living, and hospital/acute-care to name a few. The activities that nurses perform often directly impact the health and well-being of patients [5]. As a result, there is a great need for understanding the daily activities of nurses so that further links between quality nursing behavior and positive patient outcomes can be reinforced, and instances of negligent or substandard care can be identified and addressed. However, due to the complexity of nurse care activities, high data collection cost, and the lack of availability of labeled activity information, sensing-based nurse care activity recognition has not been extensively explored [10].

Outside of the nurse activity domain, sensing and tracking human activity is a highly studied research area [7]. Existing approaches include vision-based [6], Radio Frequency(RF)/WiFi-based

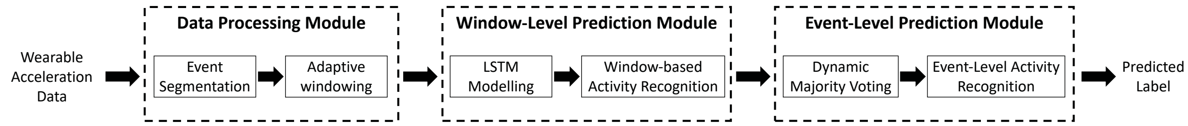


**Figure 1: Flowchart of the introduced hierarchical sequence-to-one approach for nurse care activity recognition.**

[17], structural vibration sensing [3, 8, 12, 13] and mobile/wearable-based sensing [1, 14]. These approaches rely on the insight that various human activities have distinct motion associated with them that can be used for uniquely identifying each activity. In many cases, these sensing modalities are restricted due to requirements of line-of-sight (vision), perceived privacy concerns (vision), and requiring motion through a specific path/sensing area (RF/WiFi). For nurse activity recognition, wearable-based sensing approaches provide significant benefits over many of these existing approaches due to their ubiquity, ease of instrumentation, and that they do not require any special action from the nursing staff (beyond putting on the device each day). As a result, this work focuses on leveraging acceleration-based wearable sensing for nurse activity recognition.

A common approach for wearable-based activity recognition is sequence-to-one Long Short Term Memory (LSTM). The main idea behind this approach is to take a sequence of signal windows as the input and learn a label (i.e., the activity) for the whole sequence. Although this approach has provided strong performance in various signal-based classification tasks, it faces two main challenges for nurse care activity recognition. Specifically, 1) the distribution of nursing activity instances across the known nursing activity types is imbalanced, resulting in poor predictive performance (specifically for the minority classes), and 2) there are various sources of uncertainty, such as environmental factors and noise, which makes it difficult to accurately identify activities using short duration signal windows.

To overcome the aforementioned challenges, we introduce a window-based sequence-to-one approach with dynamic voting for nurse care activity recognition. To provide the signal windows, we first separate long time-series signals into event-wise signals by detecting variance changes of the signal, and apply a sliding window to produce constant-length signal windows. To address the first challenge, we design an adaptive algorithm which determines the overlapping ratio of the sliding windows for each event based on their duration. Specifically, we use a large overlapping ratio between windows for short events and avoids overlapping for long events, which produces a more balanced distribution across the known nursing activity types. To address the second challenge, we apply a dynamic majority voting algorithm for complementing the event-wise predictions across the signal windows and reducing the effect of the uncertainties.

To evaluate our work, we showcase its performance in "The 2nd Nurse Care Activity Recognition Challenge Using Lab and Field Data" as team HealthyVibes and applied our approach to the Heiseikai data, a nurse care activity dataset [11][15] collected from real-world field experiments. With this dataset, our approach achieves 97.4% accuracy on training and 43.9% accuracy and 0.449 F1-score on validation.

## 2 SEQUENCE-TO-ONE APPROACH FOR NURSE CARE ACTIVITY RECOGNITION

In this section, we introduce our window-based sequence-to-one approach with dynamic voting for nurse care activity recognition. As shown in Figure 1, our approach contains three modules: (1) a data processing module, (2) a window-level prediction module and (3) an event-level prediction module.

### 2.1 Data Processing Module

In this module, we process the wearable acceleration data to prepare a balanced dataset of the time-domain signal windows and their corresponding activities. To this end, we first segment continuous time series signals into "events". Then, we use an adaptive sliding window to split the signals into signal windows and compile a relatively balanced dataset.

*2.1.1 Event Segmentation.* In this section, we segment the signal into events, which are potentially comprised of several signal windows [16]. The main intuition behind this step is that the occurrence of nursing activities will result in larger variance in the signal. Therefore, we identify these events by tracking the moving variance of the acceleration signal and comparing it to the background ambient vibration (when there is no event). Specifically, we first compute the L2-norm of the signals and noises from three directions (i.e., x, y, z) and then calculate the 10 second moving variance along the L-2 norm of the continuous signal. After determining the moving variance, we identify an event when the signal variance is more than three standard deviations from the variance of the ambient noise. With the "events" identified from the dataset, we conduct data interpolation to achieve a uniform sampling rate of 10Hz for each event based on the empirical observation that the number of effective data points per second is around 10 after data de-duplication. Further, we have observed that the duration distribution of the events is skewed. This means that the majority of the events have lengths of around 20s to 50s, with outliers that are shorter than 5s or longer than 15 minutes. We hypothesized that these extremely short or long events are caused by spurious factors and hence, have filtered them out and only kept the activities within the middle 80 percentile duration range for further analysis.

*2.1.2 Adaptive Windowing.* Although removing the outlier events controls the range of event duration, the event dataset is not suitable for use as a model input because of the variable window length. Thus, we use a sliding window to assure signal inputs to the model have the same length, which will reduce the model complexity and retain model generalizability. However, the label distribution is imbalanced over all the windows, which poses challenges in model training and prediction. Therefore, we designed an adaptive windowing algorithm to estimate the overlapping ratio for each event. The process for this adaptive algorithm is as follows: in the first pass, we determine the event duration by reading through all the timestamps in each event; in the second pass, we calculate the overlapping ratio as a function of event duration. This function satisfies two conditions: 1) for short events, the overlapping ratio is large between the windows, while it is small for long events; 2) for events exceeding 2 minutes, the overlapping ratio is automatically set to zero. As a result of this algorithm, the number of windows for each event label is more balanced, which achieves 2× reduction of relative ratio between the majority and minority counts (i.e., from 12 to 6). To select the best length for the sliding window, we conduct a grid search across multiple window durations and select 10 seconds because it gives the best activity prediction accuracy.

## 2.2 Window-Level Prediction Module

In this module, we first train a sequence-to-one Long Short-Term Memory (LSTM) RNN model. This model, in turn, takes each sliding window signal as a sequence input and performs window-level activity recognition.

LSTM is a commonly-used RNN architecture that was specifically designed to overcome the vanishing gradient problem encountered by traditional RNNs [2]. It maintains a memory over time and learns when to write and reset memory, and when to read information from memory. Details of the LSTM model can be found in work by Hochreiter, et al. [9].

In addition, to learn high-level features from real-world data that has large size, and to obtain a good prediction accuracy, we use a deep-LSTM that stacks multiple LSTM models on top of each other. We then add multiple fully-connected layers with nonlinear activation functions on top of the last hidden state from the last LSTM layer and apply a softmax to obtain a vector of class probabilities for each sliding window signal. The model is trained by back-propagation aiming at minimizing the cross-entropy loss between the predicted activity label and the ground truth label. Details of the architecture, hyper-parameters and training strategies for obtaining the result and preventing over-fitting is presented in Section 3.2.

## 2.3 Event-Level Prediction Module

In this module, we first perform dynamic majority voting and then use the voting results for event-based activity recognition. As mentioned before, every event contains multiple windows. Therefore, within one specific event, there may be multiple predicted activities across each of the signal windows. This likely occurs due to the 'noisy' windows that contain partial information of an event, and/or possible intervals within an event that are not relevant to that event. For instance, an event of "helping patient movements" might consist of standing still and moving the patient; "standing still" might be an activity in one window of that event but it alone it does not characterize the activity of "helping patient movements". As a result, the "standing still" window adds "noise" to the event prediction. To eliminate these 'noisy' windows, we implement a dynamic voting algorithm based on the Boyer–Moore majority vote with minor modifications [4].

The Boyer-Moore majority vote is an algorithm designed to find a majority member among a streaming sequence of labels. In our case, for example, the algorithm provides dynamic voting among the label sequences from the window-level predictions to get the event-level activities. The algorithm initializes a majority counter with zero. When a new label is entered, the algorithm checks the value of the counter: 1) if the counter is 0, this label will be stored and counter will be set to 1. 2) If the counter is non-zero, there are two cases: i) if the new label is equal to the stored label, counter is incremented by 1. ii) Otherwise, the counter is decremented by 1. This algorithm is effective at eliminating scattered minority labels among a sequence of majority labels (i.e. the predicted label with highest count). For a window sequence that is dynamically updating, since enough alternative labels should be input before the majority label count goes down, this algorithm will suffer from a time lag. Time lag on the correct label sequence will unintentionally alter the labels that are originally correct. To solve this time lag, the assignment of $i = i - 1$ can be altered to $i = max(0, i - 2)$, which will fasten the subtracting process and hence has less time lag for a new label to become the majority. By using the dynamic majority vote algorithm, the original label sequence of window-level predictions will have a continuous sequence of majority labels and scattered minority labels will be eliminated.

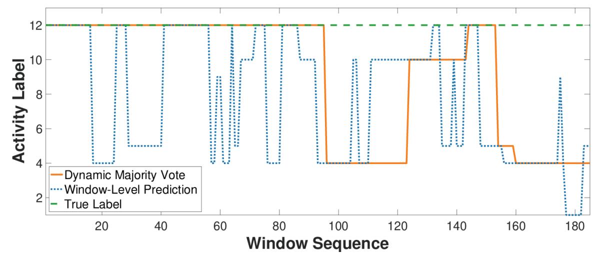


**Figure 2: The effect of dynamic majority voting on one event. Comparison of the activity labels is given by 1) the ground truth, 2) window-level prediction and 3) dynamic majority vote**

As is shown in figure 2, the solid orange line represents dynamic majority vote on the label sequence of window-level prediction from one event. From timestamp 1 to 90 of the window-level predicted sequence, minority labels are scattered among majority labels (label 12 in this case). By implementing a dynamic majority vote, scattered minority labels are altered to a majority label (which is also the true label here), thereby increasing the accuracy of the window-level prediction sequence.

# 3 EXPERIMENTAL RESULTS

In this section, we evaluate our approach on an open-access dataset collected in a care facility in Japan. In this section, we first describe the nurse care activity dataset and then discuss the evaluation results.

## 3.1 Nurse Care Activity Recognition Dataset

We first describe the nurse care activity recognition dataset [11][15]. The dataset consists of 12 nursing activities that can be categorized into 3 principle types: A) Help in Mobility, B) Assistance in Transfer, and C) Position Change. From these three principal categories, 12 activities are considered: A1) Guide (from the front), A2) Partial Assistance, A3) Walker, A4) Wheelchair, B5) All Assistance, B6) Partial Assistance (from the front), B7) Partial Assistance (from the side), B8) Partial Assistance (from the back), C9) To Supine Position or To Right Lying Position, C10) To Left Lying Position, C11) Lower Body Lifting, and C12) Horizontal Movement. The raw data was collected using accelerometers in mobile phones. The mobile phone is secured on the right arm using an armband for each nurse. The data was acquired with a 60Hz sampling rate. Due to the synchronization delay in the mobile phone, the actual sampling rate of the recorded signal varies from 1Hz to 100Hz with duplicated data points.

Two datasets are provided for training: "Lab" and "Field", which consist of accelerometer data from two users and six users, respectively. Lab data was collected in the experimental lab: the Smart Life Care Unit of the Kyushu Institute of Technology in Japan, and real field data was collected in a care facility in Japan. These two datasets are provided for "The 2nd Nurse Care Activity Recognition Challenge Using Lab and Field Data". We combine both the "Lab" and "Field" datasets into one dataset and use this combined dataset for training and validation of our models.

Each training dataset contains raw data and the corresponding activity labels. Raw data consists of User ID, timestamp, and accelerometer readings. Data labels contains a User ID and time intervals for each activity (start and end timestamp and associated activity). A matching of the raw data with labels was performed to tag specific activity to each accelerometer reading. In general, for detecting the events, we can use a variance-based approach, as discussed in Section 2.1.1. However, as the training dataset for this challenge is discontinuous, we define one "event" as the acceleration data between the start and end time of a given label. Then, we assign a label to the event using the corresponding ground truth information.

The activity labels corresponding to each window in the label data were manually entered by each nurse. As the experiments in the "Lab" data were carried out in a controlled setting, there is an activity label corresponding to every event. That is not the case for the "Field" data, where manual entry of activity in the real field setting has rendered many events without an activity label. The duration of most activities is less than 180 seconds (with the majority of them ranging from 10 to 40 seconds).

Due to the controlled experiment setup for Lab data, the count of events corresponding to each activity in the label data is almost equally distributed (minimum count is 33, maximum count is 39). But this is not the case for the "Field" data that has a wide variation of the number of events captured for each activity label (minimum Count is 1, maximum count is 423). The "Lab" data provided the base for initial exploration of data due to its uniform distribution. Also, the inclusion of the "Lab" data in the training set increases number of observation of activities 2,8,11 that are poorly represented in the "Field" data.

The testing data was collected in field setting from three nurses. The raw testing data provided consists of accelerometer data and timestamp information for each User ID.

It is expected that the accelerometer signal pattern varies for each activity. This is corroborated in Figure 3 where a stark difference of signal types can be observed between Activity 4 and 5 for the

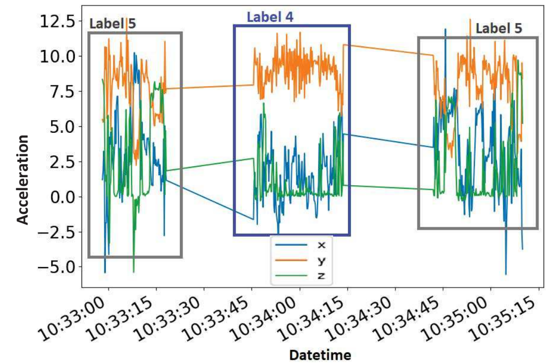


**Figure 3: A visualization of one section of data to show discontinuity between two activities of user 4. A change in the signal pattern is noted as the user performs different activities. The activity visualised involves help in mobility on a wheelchair (Label 4) and assistance in transfer (Label 5)**

same user. It is also noted that the same activity can have different signal patterns for different users.

## 3.2 Evaluation Results

In this section, we evaluate the performance of our method with the pre-mentioned open-access dataset. First, we tune the model parameters, select training strategies, and evaluate the overall model performance based on the training and validation accuracy. Then, we compare the model performance with and without adaptive windowing to demonstrate the effectiveness of this method in performance improvement. Finally, we evaluate the effectiveness of the dynamic voting by comparing the model accuracy with and without it.

*3.2.1 Overall Performance.* The model architecture we selected in this work is the deep-LSTM, where we stack multiple layers of LSTM sequence together to form a deep LSTM model. The input of the model are windows of 10 seconds with tri-axial accelerations (i.e., a 3-d tensor of shape (#windows, 100, 3)). We split our training and validation set at an event-level. We randomly split events into two sets (with a split ratio of 0.15) and then assign all windows within each event to either the training or validation set. To improve the quality of the features, we tune the data-processing parameters such as window size, window overlapping ratio, etc. After a grid search, we use 10 seconds as our window size, which gives the best prediction accuracy. To determine the overlapping ratio between windows, we implement the adaptive algorithm that calculates the overlapping ratio based on the event duration, which enables shorter events to generate more windows. This will be further discussed in Section 3.2.2. Also, we conduct iterative label down-sampling at each training epoch, which allow us to reduce over-fitting towards majority labels without losing any information from the complete training data. After that, we conduct a grid search over various combinations of the hyper-parameters, including the learning rate, the number of LSTM layers, hidden dimensions, regularization coefficient, etc. to determine the best set of hyper-parameters.

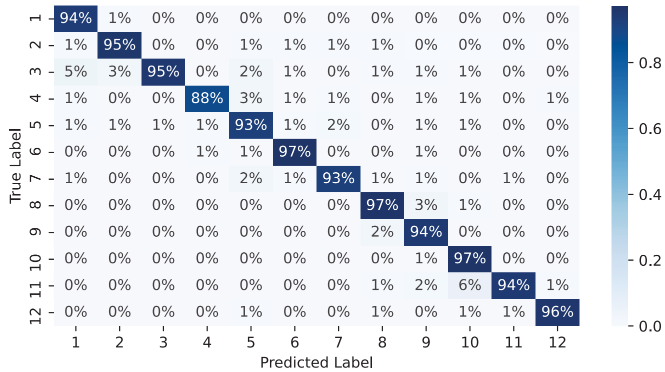


**Figure 4: Confusion matrix for the training set with accuracy of 97.4%. The model has large model capacity for activity prediction.**

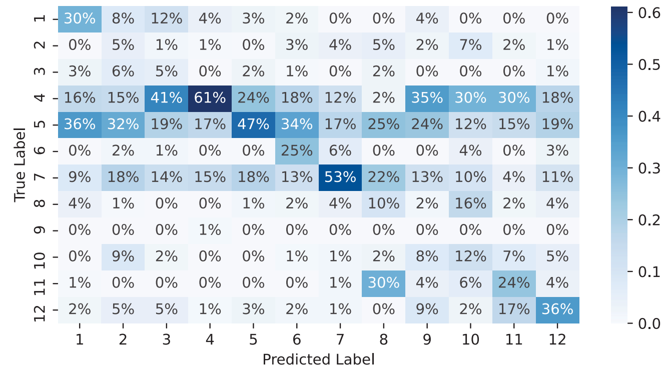


**Figure 5: Confusion matrix for the validation set with accuracy and F1-score of 43.9% and 0.449, respectively. The differences in training and testing indicate that the model is over-fitting towards the training dataset, and there are large differences between the events.**

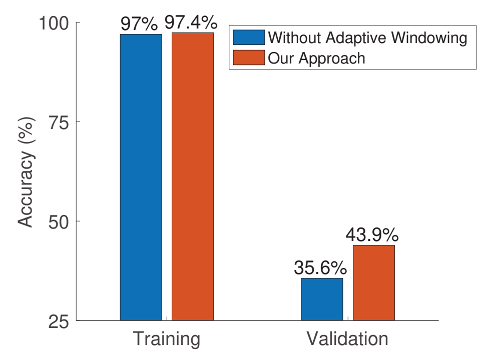


**Figure 6: Comparison between our approach and the approach without adaptive windowing.**

To improve the training efficiency, we use the Adam optimizer, which incorporates gradient decaying and momentum to avoid gradient explosion/vanishing and pass the local minima/saddle points. In addition, we use early stopping to monitor the gradient updates and cut-off the training process when the loss stops decreasing. After obtaining the predictions from the model, we conduct dynamic voting over the entire sequence of each event to compute the final prediction results. This will be discussed in Section 3.2.3.

The metrics we used for evaluation are accuracy and F1-score because of the imbalanced label distribution in the validation set. The overall training accuracy is 97.4% and the validation accuracy is 43.9% and 0.449 for F1-score. The training accuracy is high because the deep-LSTM model has sufficient capacity to incorporate variances in the dataset. However, it overfits to the dataset and gradually loses generalizability because of the differences between events. The discrepancy between the training and validation performance indicates that the event-level differences are large. Based on our definition of an "event", these differences include time, location, service target, nursing style, etc., which requires a larger training set to incorporate all these factors. The confusion matrices of training and validation set are shown in Figure 4 and 5, respectively. We observe that the majority of the wrong predictions are on activity 4 and 5, indicating "wheelchair assitance" and "all assistance in transfer" are more likely to be mixed up with the other activities. This is because both activities contain similar motions as the other activities: Activity 5 includes all assistance in transfer movements, which overlaps with singular partial assistance. Activity 4 is similar to walking because wheelchair assistance includes walking when pushing the wheelchair.

*3.2.2 Effectiveness of Adaptive Windowing.* Adaptive windowing enables different length of "event" to have similar window numbers, which leads to more balanced label distribution for windows-level prediction. We compare the accuracy of our activity prediction approach (which uses the adaptive windowing) and the approach without adaptive windowing. Prior to applying these algorithms (left bar), the model is trained on a imbalanced dataset, which overfits to the majority activity labels. After the adaptive windowing (right bar), the model is trained on a balanced dataset for each epoch, so that all the activity types have the same weight when computing the loss in each iteration.

As observed in Figure 6, the training accuracy does not change much because in both models tends to overfit to its training data. In contrast, the model performance in validation improves significantly by 8.3% after the adaptive windowing. It is because our model, which is trained with balanced dataset, captures more features between different activity types since they contributes equally to the loss.

*3.2.3 Effectiveness of Dynamic Majority Voting.* To improve the accuracy by eliminating "noisy" windows among window sequences of unique labels, the Boyer–Moore majority vote algorithm was implemented and evaluated. Figure 7 shows the result of our approach which uses this majority vote and compares it with initial inter-window prediction from LSTM network (i.e., before voting). For training accuracy, our approach increased the accuracy by 1.9%, due to the elimination of noise window label modification. While for validation, our approach improved the accuracy by 9.9%. This improvement is primarily because dynamic majority voting altered

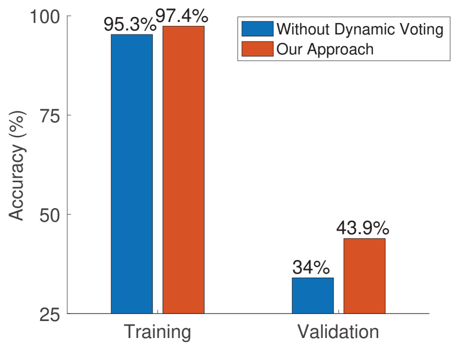


**Figure 7: Comparison between our approach and the approach without dynamic voting.**

"noisy" labels into their adjacent majority labels, which makes the labels within one event less scattered (as shown in Figure 2). Therefore, the accuracy improves as the number of true labels increases.

## 4 CONCLUSION

In this paper, we present a window-based sequence-to-one approach with dynamic voting for nurse care activity recognition using acceleration-based wearable sensors. This approach enables classification of nursing activity using wearable accelerometer-based sensing despite overlapping signals and similar body movements across varying activities. We developed: 1) an adaptive windowing algorithm to balance the dataset by estimating the overlapping ratio for event signals with different windows, and 2) a majority voting algorithm to determine the majority prediction of an event signal that contains a sequence of sliding window signals. We evaluate our approach on an open-access dataset that consists of 6 nurses and 12 nursing activities. Our approach achieves 97.4% training accuracy and 43.9% validation accuracy. The recognition result for the testing dataset will be presented in the summary paper of the challenge [15].

## ACKNOWLEDGMENTS

This research was supported in part by Highmark, the National Science Foundation (under grant CMMI-1653550). The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of Stanford, CMU, UCM, NSF, or the U.S. Government or any of its agencies.

## REFERENCES

[1] Bashar Altakouri, Gerd Kortuem, Agnes Grünerbl, Kai Kunze, and Paul Lukowicz. 2010. The benefit of activity recognition for mobile phone based nursing documentation: A wizard-of-oz study. In *International Symposium on Wearable Computers (ISWC) 2010*. IEEE, 1–4.

[2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[3] Amelie Bonde, Shijia Pan, Hae Young Noh, and Pei Zhang. 2019. Deskbuddy: an office activity detection system: demo abstract. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. 352–353.

[4] Robert S Boyer and J Strother Moore. 1991. MJRTY—a fast majority vote algorithm. In *Automated Reasoning*. Springer, 105–117.

[5] Dorothy Brooten and Mary D Naylor. 1995. Nurses' effect on changing patient outcomes. *Image: The Journal of Nursing Scholarship* 27, 2 (1995), 95–99.

[6] Claudette Cedras and Mubarak Shah. 1995. Motion-based recognition a survey. *Image and Vision Computing* 13, 2 (1995), 129–155.

[7] Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu. 2012. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 6 (2012), 790–808.

[8] Jonathon Fagert, Mostafa Mirshekari, Shijia Pan, Pei Zhang, and Hae Young Noh. 2017. Monitoring hand-washing practices using structural vibrations. *Structural Health Monitoring* (2017).

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Md Eusha Kadir, Pritom Saha Akash, Sadia Sharmin, Amin Ahsan Ali, and Mohammad Shoyaib. 2019. Can a simple approach identify complex nurse care activity?. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. 736–740.

[11] Sozo Inoue; Sayeda Shamma Alia; Paula Lago. 2020. Nurse Care Activities Datasets: In laboratory and in real field. https://doi.org/10.21227/jem3-ap07

[12] Mostafa Mirshekari, Jonathon Fagert, Shijia Pan, Pei Zhang, and Hae Young Noh. 2020. Step-Level Occupant Detection across Different Structures through Footstep-Induced Floor Vibration Using Model Transfer. *Journal of Engineering Mechanics* 146, 3 (2020), 04019137.

[13] Mostafa Mirshekari, Shijia Pan, Jonathon Fagert, Eve M Schooler, Pei Zhang, and Hae Young Noh. 2018. Occupant localization using footstep-induced structural vibration. *Mechanical Systems and Signal Processing* 112 (2018), 77–97.

[14] Ren Ohmura, Futoshi Naya, Haruo Noma, and Kiyoshi Kogure. 2006. B-pack: a bluetooth-based wearable sensing device for nursing activity recognition. In *2006 1st international symposium on wireless pervasive computing*. IEEE, 6–pp.

[15] Sayeda Shamma Alia; Paula Lago; Kohei Adachi; Tahera Hossain; Hiroki Goto; Tsuyoshi Okita and Sozo Inoue. 2020. Summary of the 2nd Nurse Care Activity Recognition Challenge Using Lab and Field Data.

[16] Shijia Pan, Mario Berges, Juleen Rodakowski, Pei Zhang, and Hae Young Noh. 2019. Fine-grained recognition of activities of daily living through structural vibration and electrical sensing. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 149–158.

[17] Yoshihiro Takebe, Masako Kanai-Pak, Noriaki Kuwahara, Jukai Maeda, Miwa Hirata, Yasuko Kitajima, and Jun Ota. 2013. Recognition of nursing activity with accelerometers and RFID. *Kybernetes* (2013).

## APPENDIX

Table 1 summarizes the pipeline and resources for our solution. In post-processing, we transform all the window-level predictions to the timestamp-level labels. First, we obtain the start and end time of each window. For each window, timestamps from raw data that are within the start and end time are assigned with corresponding label of that window. By iterating every predicted window, labels with predictions are mapped into the raw data file. Finally, we obtain a complete table with user ID, timestamp and label.

**Table 1: Description of the pipeline**

| | |
|---|---|
| Sensor modalities | Accelerometer |
| Features | Raw tri-axial acceleration signals |
| Programming information | Python, Keras, numpy, sklearn |
| Window size | 10s |
| Training time | 1200s |
| Training Machine | Tesla K80 GPU (24GB RAM) |
| Testing time | 10s |
| Testing Machine | Macbook Pro CPU (16GB RAM) |