AN AUTOENCODER-BASED APPROACH FOR RECOGNIZING NULL CLASS IN ACTIVITIES OF DAILY LIVING IN-THE-WILD VIA WEARABLE MOTION SENSORS

Ali Akbari[1] , *Roozbeh Jafari*[1,2,3]

[1] Department of Biomedical Engineering, Texas A&M University
[2] Department of Computer Science and Engineering, Texas A&M University
[3] Department of Electrical and Computer Engineering, Texas A&M University

ABSTRACT

Recognizing activities of daily living (ADL) in-the-wild, while users follow their daily routine, is challenging due to the presence of various activities that do not belong to the set of desired activities in which the system is interested (*i.e.*, NULL class). In this paper, we propose a framework for ADL recognition via wearable motion sensors with the ability to detect NULL class. Existing ADL recognition systems either ignore the NULL class or use some training data to train a model for recognizing it. However, our framework uses only samples of the desired activities in the training phase and learns to detect the NULL samples based on a modified variational autoencoder model that outputs reconstruction probability. Experimental results show that in detecting six ADL with accelerometer data, our system achieves 14% higher F1-score compared to the models that use training samples of NULL activities.

Index Terms— Wearable motion sensor, ADL recognition, variational autoencoder, NULL class detection

1. INTRODUCTION

Recognizing activities of daily living (ADL) with motion sensors is gaining a bold traction in mobile computing as it provides vital information about people and their activities which can enhance the effectiveness of many real-world applications [1]. In contrast to controlled data collection paradigm, in real-world scenarios, only a few parts of a sensor data are relevant for ADL recognition systems, and a big portion of the motion data usually belongs to activities for which the system is not trained. Those irrelevant activities, called NULL activities, introduce a big challenge for ADL recognition systems as the system does not know how to deal with them. Our aim is to develop a framework for recognizing ADL in-the-wild using wearable motion sensors with the ability to distinguish between NULL and desired activities by only using the samples of the desired activities in the training phase.

This is significant because the NULL class represents a theoretically infinite space of arbitrary activities and thus explicitly modeling of it is very challenging, if not impossible. Moreover, contrary to the systems that are trained for controlled environments, real-world ADL recognition systems cannot ignore samples of the NULL class since those samples usually constitute a major portion of the datasets. Such a system can monitor the activities under realistic, everyday life conditions, which can be useful in a wide variety of applications, such as health monitoring [2].

To avoid the issues with NULL activities, existing ADL recognition systems either restrict the experiments by asking participants to perform only the activities of interest and nothing else, or manually remove the samples of NULL activities from their analysis. The performance of these systems degrades drastically when the system is tested in-the-wild due to the natural occurrence of countless NULL activities, especially when these resemble the ADLs of interest. Several systems have attempted resolving this degradation by defining an extra NULL class during training. These systems still suffer from the important problem that they can only handle NULL activities that were given during the training phase (*i.e.*, previously unseen activities can be confusing to the system).

To address the aforementioned issues in real-world ADL recognition, we propose a two-stage activity recognition system that is able to recognize the samples of the NULL activities only based on using samples of desired activities in training. In the first stage, the system tries to detect if an input data belongs to the set of desired activities on which the system is initially trained. If not, the system will ignore that sample. We approach this as a NULL class detection problem where we modify a variational autoencoder (VAE) to provide a probabilistic reconstruction error, instead of the subjective absolute error, for detecting NULL samples. We also propose a measure for understanding the confidence of the classifier based on which the system can solicit the user for providing annotations when it is not confident about an input. In summary, the contributions of this paper are as follows:

- We propose an in-the-wild ADL recognition system based on a VAE that can recognize NULL activities without needing their samples in the training.
- We modify a typical VAE to output the probability of

reconstruction instead of the absolute error, which allows to have a global criteria for distinguishing between NULL and desired activities.

- We show the effectiveness of our algorithm through several experiments with real-world smartphone's data.

2. BACKGROUND

Many studies have tried to perform ADL recognition in-the-wild by using wearable motion sensors, but very few of these have elimination of NULL activities as their focus [3–7]. One approach estimates the confidence of a classifier to detect NULL samples at the end of classification, but this has a high false-positive rate as it misclassifies ADLs with a low classifier confidence [8]. The samples from all NULL activities are used to train a separate NULL class in addition to the desired activities [1]. The performance of this system degrades when it encounters novel samples, which are not included in the set of NULL activities that are used in the training phase.

In applications other than ADL recognition, however, several NULL class detection techniques have been proposed to detect samples that do not belong to the initial training data. Distance-based approaches put a threshold on the distance between the new data point and normal data to determine whether it is a NULL sample [9, 10]. The threshold is tuned subjectively, and it should be reconfigured when the structure of the data is changed. Probabilistic approaches estimate the probability density function of the input data and threshold it to define the boundaries of normal data [11, 12]. They require complex algorithms to estimate the true distribution of the data. Reconstruction-based algorithms try to recreate the inputs through which they also learn the structure of the dataset. They compare reconstruction error, which is the difference between the output of the system and the original input, to a constant threshold to detect the NULL class [13–15].

3. METHODS

Detecting instances of NULL activities is essential for in-the-wild ADL recognition systems. To distinguish between the NULL and desired activities' samples, we propose a two-stage framework as shown in Fig. 1. In the first stage, we train a VAE only on the samples of the desired activities [16]. The reconstruction probability is then used for distinguishing between NULL and desired activities. In the second stage, the samples that are detected as NULL are removed. For the remaining samples, the features created by the VAE are fed to a classifier for recognizing the activity. The system is also able to estimate its confidence which can be further used to remove samples that might belong to the NULL class or ask for user's feedback when it is not certain about an input.

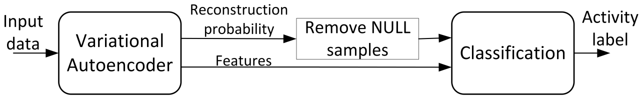

Fig. 1. An overview of the proposed system for ADL recognition in-the-wild

3.1. NULL class detection

3.1.1. Variational autoencoder

Autoencoder (AE) is a type of neural network that can learn the intrinsic structure of the data in an unsupervised manner. This is done by mapping a high-dimensional input (x) to a low-dimensional latent space (z) and then reconstructing the input from the latent space. The first part is called the encoder ($g : x \to z$) and the second part is decoder ($f : z \to x$). Through this process, the AE ($f \circ g : x \to x$) learns the structure of the low-dimensional manifold that the data lies on. Reconstruction error, which is the difference between the original input and its reconstructed copy (*i.e.*, output of the AE), is equivalent to distance from the input to the manifold.

VAE is a specific type of AE that treats the latent variable (z) as a random variable by assigning a Gaussian probability distribution to it [17]. Thus, VAE is a useful tool for modeling uncertainty in motion signals that come from user variations and sensor noise. The objective function given in Equation 1 is used to train a VAE:

$$\mathcal{L}(x) = -\log p(x|z) + D_{kl}\{q(z|x)||p(z)\} \qquad (1)$$

where 1, $q(z|x)$ is posterior distribution of the latent variable z estimated by the encoder. $p(z)$ is the prior for the latent variable and is chosen as a zero-mean, unit-variance Gaussian. The output of the encoder is mean and standard deviation of a Gaussian that serve as the parameters of $q(z|x)$. Finally, the $-\log p(x|z)$ is the reconstruction error which is estimated by the decoder. This can be obtained by minimizing conventional loss functions such as mean squared error.

In a deep VAE, each of encoder and decoder is created by stacking multiple hidden layers. We choose three convolutional hidden layers for encoder (with 32 neurons in each layer), which can extract informative features from the raw data automatically [3]. Three deconvolutional layers are also used for the decoder. Based on our experiments, it was difficult to obtain a reasonable accuracy with a lesser number of layers. Increasing the layers, however, increases complexity of the model and makes it difficult to be run on wearable devices [18], without significantly improving the performance. Our data stream of the 3D accelerometer is segmented into 3-second windows with an overlap of 50%, and each window is fed as one input to the neural network.

3.1.2. NULL class detection

The reconstruction error of the VAE would be small for any data created by the same distribution as the training data.

Contrarily, data that is not created by a similar distribution as the training data (do not lie on the same low-dimensional manifold) would have a large reconstruction error [15]. Therefore, by comparing the reconstruction error of the VAE with a constant threshold, we can detect NULL samples that are not similar to the training data. However, as the absolute reconstruction error ($|x - f \circ g(x)|$) depends on the values of the data, such a threshold would be subjective, and it should be changed when the subject or the configuration of the sensors changes. To address this issue, we modify the typical VAE to output the reconstruction probability instead of the absolute error. The likelihood $p(x|z)$ for one data point is modeled with a Gaussian distribution as Equation 2.

$$p(x|z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-f(z))^2}{2\sigma^2}} \tag{2}$$

where σ^2 is the variance and $f(z)$ is the output of the VAE. The negative log likelihood can then be written as Equation 3.

$$-\log p(x|z) = \frac{1}{2\sigma^2}(x - f(z))^2 + \log \sigma\sqrt{2\pi} \tag{3}$$

Typically in VAE models the σ is assumed to be constant so minimizing the negative log likelihood becomes equivalent to minimizing squared error loss function (the first term on the RHS of Equation 3). To retrieve the reconstruction probability, which is a general metric and does not depend on the dataset, we propose to model the σ as an output of the neural network and try to estimate its value for each input data. Thus, the decoder in our framework has two outputs. One is the reconstructed version of input x (*i.e.*, $f(z)$) and the other is the variance of this estimation. By estimating the variance in addition to the typical output of the VAE, we can calculate the actual reconstruction probability by Equation 3. This value is higher for the samples similar to the training data and lower for new samples such as NULL class samples. Note that the probability is a generic metric that is independent of the data. The samples for which the reconstruction probability is lower than a threshold (we set it to 0.6 based on our experiments) are assumed to be from the NULL class and are removed. The samples with reconstruction probability higher than the threshold are fed to the classifier to detect the activity.

3.2. ADL recognition

The latent variable z created by the encoder in the VAE is a compressed version of the input data that contains the most informative features of it. These features can be leveraged for recognizing the ADL, which removes the need for extracting manual features from the raw data. In fact, the raw data is fed to the convolutional layers of the VAE and the features are extracted automatically. We use three fully connected layers for combining the features and mapping them to the ADL. The first two contain 32 neurons, and the last one contains the same number of the neurons as the number of desired ADL.

The outputs of the encoder are the parameters of a Gaussian distribution (*i.e.*, mean and standard deviation) that serve as the posterior distribution of the features, given a data point. For each single input datum x_i, we sample from this distribution for $N = 50$ times and feed all those samples to the classifier network. The output of the classifier for all N samples is calculated and the Monte-Carlo estimation of the mean of the outputs is taken as the final decision of the classifier.

$$\bar{c} = \frac{1}{N}\sum_{j=1}^{N} h(z_j) \qquad z_j \sim \mathcal{N}(\mu_i, \sigma_i) \tag{4}$$

where h is the classification network, μ_i and σ_i are the outputs of the encoder for i^{th} input data, and $\bar{c}$ is the final decision of the classifier which determines the activity of interest.

We can also leverage empirical standard deviation of the outputs of the classifier for N samples (Equation 5) to estimate the confidence of the classifier. For the samples that the classifier is confident about, the labels would be more consistent, while for non-confident ones, the classifier would generate distinct labels that leads to a higher standard deviation.

$$s = \Big(\frac{1}{N}\sum_{j=1}^{N}(h(z_j) - \bar{c})^2\Big)^{\frac{1}{2}} \qquad z_j \sim \mathcal{N}(\mu_i, \sigma_i) \tag{5}$$

4. EXPERIMENTAL RESULTS AND DISCUSSION

To demonstrate the performance of our framework we use a publicly available ADL dataset called Actitracker that contains real-world ADL data captured by a smartphone from multiple subjects [19]. We use the data from first 10 subjects. The labeled data includes sitting, standing, lying down, walking, jogging, and stair climbing and we call them basic ADLs. In addition to the labeled data of those six basic ADLs, this dataset contains a large amount of unlabeled samples form other activities that users performed during their normal daily living. Unlabeled data, which constitutes the NULL class, has 13 times as many samples as the whole labeled data.

4.1. NULL and ADL recognition performance

We first investigate the performance of our method in detecting NULL form basic ADLs, described in Section 3.1.2. We consider all basic ADLs as the non-NULL class to evaluate performance of the models regarding distinguishing between NULL and non-Null (a two-class classification). Table 1 represents the F1-score in NULL class detection with 10-fold and leave-one-subject-out (LOSO) cross-validation and compares it to other methods of NULL detection described in [1]. The ”bgClass” defines a NULL class in addition to the basic ADLs, ”preReject” inserts a two-class classifier before actual ADL classification step, and ”postReject” adds the NULL rejection step after classifying the ADLs (for more details see [1]). We applied those NULL detection approaches

Table 1. Average F1-score of NULL class detection

	10-Fold	**LOSO**
bgClass [1]	74%	67%
preRejcet [1]	77%	70%
postReject [1]	76%	70%
Our method	**92%**	**85%**

to a classifier with a similar structure as ours (Section 3.2) to have a fair comparison. It should be noted that all these methods use some samples of the NULL class in their training phase, which is the nowadays common approach for dealing with NULL activities. We chose 50% of the NULL samples randomly to train models in [1] and the remaining 50% was used in the testing phase. In contrast to the models in [1], our model does not use any training data from the NULL class.

As Table 1 shows, our system outperforms the models in [1] by 15% in both 10-fold and LOSO validations regarding NULL detection task, exhibiting its superior ability in detecting and removing NULL samples. This superiority is due to the fact that the models that use the training data from the NULL class (models in [1]) are only capable of recognizing a limited set of NULL activities (those that are available in the training). However, the NULL class could include a wide variety of unknown activities, and those systems do not know how to deal with samples of a new activity that they have never seen before. Our system, however, can detect any sample that differs from the samples of the basic activities.

We then evaluate the system in recognizing the six basic ADLs in the presence of the NULL class. In our framework, the NULL detection step removes the samples detected as NULL and passes the non-NULL samples to the classifier to recognize one out of six ADLs. In Table 2, we compare the F1-score of our framework with models in [1] regarding ADL recognition. As Table 2 shows, our method achieves higher accuracy, 14% in 10-fold and 13% in LOSO, compared to the models in [1] regarding recognizing six ADLs. In fact, the system that can better detect and remove NULL samples can obtain higher accuracy in terms of recognizing ADLs too. In addition, we investigate the performance of a baseline model that does not have any NULL detection/removal step to see how the models that are trained in controlled environments (*i.e.*, without considering NULL class) perform in-the-wild, where NULL class samples are present. For the baseline model, we train a classifier similar to our classifier (neural network with the same structure) with only six basic activities without including any NULL rejection step. According to Table 2 the baseline model fails significantly when the samples of NULL class are present in the testing phase since the amount of NULL samples in testing is much larger (6 times) than the basic ADLs. The reason is that this model does not detect and remove NULL class samples, so it confuses them with the basic ADLs. This proves the need for considering NULL class for in-the-wild ADL recognition systems.

Table 2. Average F1-score of ADL recognition in the presence of NULL class

	10-Fold	**LOSO**
Baseline	13%	10%
bgClass [1]	70%	65%
preRejcet [1]	73%	67%
postReject [1]	72%	65%
Our method	**87%**	**80%**

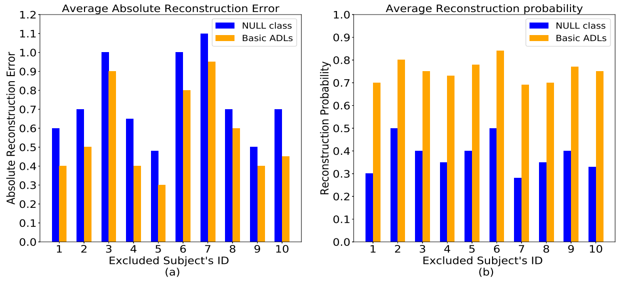

Fig. 2. Comparing average (a) absolute reconstruction error to (b) reconstruction probability for basic and NULL activities

4.2. Reconstruction probability evaluation

To study the impact of using reconstruction error on NULL class detection, we compare it to the case of using absolute reconstruction error. Figure 2 represents the mean of absolute reconstruction error (Figure 2-a) as well as reconstruction probability (Figure 2-b), proposed in this paper (Equation 3), for samples of NULL and basic ADLs (non-NULL) for 10 subjects (horizontal axis). As mentioned in Section 3.1.2, the absolute reconstruction error is a subjective measure that changes significantly when the range of the input data changes. It can be seen in Figure 2-a that this value changes significantly from one subject to another, which makes it impossible to set a global threshold that can distinguish NULL from basic ADLs for all subjects. However, Figure 2-b shows that the reconstruction probability is more consistent among different subjects, which allows us to set a global threshold for all the subjects (0.6 in our experiments).

5. CONCLUSION

We proposed a framework for recognizing ADL in-the-wild using VAE. The proposed system is capable of detecting samples of NULL activity that constitute a major part of real-world data. Our system works based on wearable motion sensors and can provide important and useful contextual information about the users and their activities and can unlock many real-world sensing and computing paradigms.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation, under grants CNS-1734039, EEC-1648451, and ECCS-1509063.

7. REFERENCES

[1] Attila Reiss, Didier Stricker, and Gustaf Hendeby, "Towards robust activity recognition for everyday life: Methods and evaluation," in *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 25–32.

[2] Yu-Jin Hong, Ig-Jae Kim, Sang Chul Ahn, and Hyoung-Gon Kim, "Mobile health monitoring system based on activity recognition using accelerometer," *Simulation Modelling Practice and Theory*, vol. 18, no. 4, pp. 446–455, 2010.

[3] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition.," in *Ijcai*, 2015, vol. 15, pp. 3995–4001.

[4] Chen Chen, Huiyan Hao, Roozbeh Jafari, and Nasser Kehtarnavaz, "Weighted fusion of depth and inertial data to improve view invariance for real-time human action recognition," in *Real-Time Image and Video Processing 2017*. International Society for Optics and Photonics, 2017, vol. 10223, p. 1022307.

[5] Johan Wannenburg and Reza Malekian, "Physical activity recognition from smartphone accelerometer data for user context awareness sensing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3142–3149, 2017.

[6] Jian Wu, Lu Sun, and Roozbeh Jafari, "A wearable system for recognizing american sign language in real-time using imu and surface emg sensors.," *IEEE J. Biomedical and Health Informatics*, vol. 20, no. 5, pp. 1281–1290, 2016.

[7] G De Leonardis, S Rosati, G Balestra, V Agostini, E Panero, L Gastaldi, and M Knaflitz, "Human activity recognition by wearable sensors: Comparison of different classifiers for real-time applications," in *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE, 2018, pp. 1–6.

[8] D Roggen, S Magnenat, M Waibel, and G Tröster, "Designing and sharing activity recognition systems across platforms: methods from wearable computing," *IEEE Robotics and Automation Magazine*, vol. 12, pp. 83–95, 2011.

[9] A Hassan, Hoda MO Mokhtar, and Osman Hegazy, "A heuristic approach for sensor network outlier detection," *Int J Res Rev Wirel Sens Netw (IJRRWSN)*, vol. 1, no. 4, 2011.

[10] Caglar Aytekin, Xingyang Ni, Francesco Cricri, and Emre Aksu, "Clustering and unsupervised anomaly detection with l2 normalized deep auto-encoder representations," *arXiv preprint arXiv:1802.00187*, 2018.

[11] Charu C Aggarwal and Philip S Yu, "Outlier detection with uncertain data," in *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, 2008, pp. 483–493.

[12] Jian Zhang, Zoubin Ghahramani, and Yiming Yang, "A probabilistic model for online document clustering with application to novelty detection," in *Advances in Neural Information Processing Systems*, 2005, pp. 1617–1624.

[13] Yuta Kawachi, Yuma Koizumi, and Noboru Harada, "Complementary set variational autoencoder for supervised anomaly detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2366–2370.

[14] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al., "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 187–196.

[15] Jinwon An and Sungzoon Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, pp. 1–18, 2015.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, http://www.deeplearningbook.org.

[17] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[18] Sourav Bhattacharya and Nicholas D Lane, "From smart to deep: Robust activity recognition on smartwatches using deep learning," in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[19] Jeffrey W Lockhart, Gary M Weiss, Jack C Xue, Shaun T Gallagher, Andrew B Grosner, and Tony T Pulickal, "Design considerations for the wisdm smart phone-based sensor mining architecture," in *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data*. ACM, 2011, pp. 25–33.