

PAPER • OPEN ACCESS

Multilayer Perceptron for Activity Recognition Using a Batteryless Wearable Sensor

To cite this article: D N S Putra and I N Yulita 2019 *IOP Conf. Ser.: Earth Environ. Sci.* **248** 012039

View the [article online](#) for updates and enhancements.

Multilayer Perceptron for Activity Recognition Using a Batteryless Wearable Sensor

D N S Putra¹ and I N Yulita¹

¹Universitas Padjadjaran, Jl. Raya Bandung-Sumedang, Km. 21 street Jatinangor, Sumedang District 45363, Indonesia

nurkurniawandwiki@gmail.com

intan.nurma@unpad.ac.id

Abstract. Smart City is one of the trends around the world today. In order to achieve a smart city environment, we need everything to be connected and to be managed. As one of the aspects that have strong impact on people life is their own activities, a smart city should be accompanied by smart people which could be actualized by using sensors to recognize their activities. In this research, we present and evaluate a method to recognize the gesture of someone leaving bed using RFID device. We use a classification approach in our system to conduct the experiment. The method that we are using is MLP (Multi-Layer Perceptron). By using this method, we got 90.17% of accuracy, which is slightly better than Naive Bayesian that got 84.46% of accuracy.

1. Introduction

Nowadays, cities are utilizing massive digital transformation to support sustainable urban communities in urban environment. Millions of homes are equipped with smart devices that could generate a massive amount of data to analyzed in order to support people daily activities to achieve this. Smart devices in other hand, are devices that used sensors to collect data and processed the data and then take action. As reported on [1][2], machine learning can be used to recognize people daily activities. The usage of machine learning techniques can help us to process the massive amount of data for providing actions [3].

The usage of sensors are not a new thing on smart cities. This include any kind of batteryless wearable sensor. Including the one that can be placed on chest with the receiver on any fixed position inside the room [4]. The wearable sensor can be placed on several parts of human body [attal]. The placement of wearable sensor can be burdensome for some people. Thus, the batteryless sensor can be used to reduce this feeling of burdensome because the sensor became lighter than a battery powered sensor. The other advantages of using batteryless sensor are the person using can move more freely. One of the batteryless sensor are W²ISP [4] that can be placed on the chest of the subject.

After collecting the data, the next step are to analyze the data and put them into a model that can recognize them. A lot of researcher has done this before. Some are mainly focused on bed and chair leaving motion [5][4] daily human activities [6][7][8]. Thus, this paper will discuss about batteryless wearable sensors usage for motion detection, mainly focused on bed and chair leaving motion. To do so, an algorithm that can differentiate the motions are needed, some of them are naïve Bayesian and MLP method. This process is called as a classification process. The MLP is a kind of machine learning



method. This method processes the data to form a model. Also, MLP has been used for a lot of cases, some of them are biometric GAIT identification [9], landslide susceptibility [10], thermal conductivity prediction [11], sleep stages. According to those report, the MLP have achieved a high accuracy rate for recognizing in a lot of cases. Because of that, we propose the usage of MLP to recognize the motion of people leaving bed and chair using a batteryless wearable sensor. The naïve Bayesian method in other hand, are still used in data mining because of the simplicity, efficiency, and the efficacy [12] The main contributions of this paper are as follows :

- This paper propose a human activity recognition model based on the gesture of waking up from bed.
- This paper applies a MLP method for activity prediction based on the sensor readings. For comparison, we also tested the data using a Naive Bayesian network.

The organization of this paper is as follows: Section 2 discusses the related work. In section 3, the proposed model is presented and the evaluation and result analysis in section 4. Finally, the conclusion of this paper and future direction discussion in section 5.

2. Method

In this section, the method used to do the experiment will be discussed. This includes the pre-processing of data and the model we built to recognize the activity we mentioned before.

2.1. Synthetic Minority Oversampling Technique (SMOTE)

The SMOTE method to normalize any imbalance class that appeared on the dataset using the over-sampling method. SMOTE method operate in “feature space” rather than “data space” [13]. Meaning that the process will be repeated as many as the minority classes. Below are the pseudo-code for the SMOTE method.

Algorithm SMOTE(N, F, n)

Input: Number of minority class samples N; Amount of SMOTE F%; Number of nearest neighbors n

Output: (F/100) * N synthetic minority class samples

(* If F is less than 100%, randomize the minority class samples. *)

2. if $F < 100$

3. Randomize the N minority class samples

4. $N = (F/100) * N$

5. $F = 100$

6. endif

7. $F = (\text{int})(N/100)$

(* The amount of SMOTE is assumed to be in integral multiples of 100. *)

8. n = Number of nearest neighbors

9. lenattr = Number of attributes

10. Sample[][]: array for original minority class samples

11. newidx: keeps a count of number of synthetic samples generated, initialized to 0

12. Synthetic[][]: array for synthetic samples

(* Compute k nearest neighbors for each minority class sample only. *)

13. for $i \leftarrow 1$ to N

14. Compute k nearest neighbors for i, and save the indices in the nnarray

15. Populate(F, i, nnarray)

16. endfor

17. Populate(F, i, nnarray)

(* Function to generate the synthetic samples. *)

18. while $F \neq 0$

Choose a random number between 1 and n, call it nn for one of the k nearest neighbors of i.

```

19. for attr ← 1 to lenattr
20.     Compute: dif = Sample[narray[n]][attr] - Sample[i][attr]
21.     Compute: gap = random number between 0 and 1
22.     Synthetic[newidx][attr] = Sample[i][attr] + gap * dif
23. endfor
24. newidx++
25. F = F - 1
26. endwhile
27. return (* End of Populate. *)

```

End of Pseudo-Code

2.2. MLP

The MLP is an artificial neural network with feed-forward architecture built with multiple layer, as the name implies. MLP is based on non-linear activation for the hidden units. This network architecture minimizes the error function between the estimated and real output (Attal et al., 2015). The architecture can be seen on the figure 1.

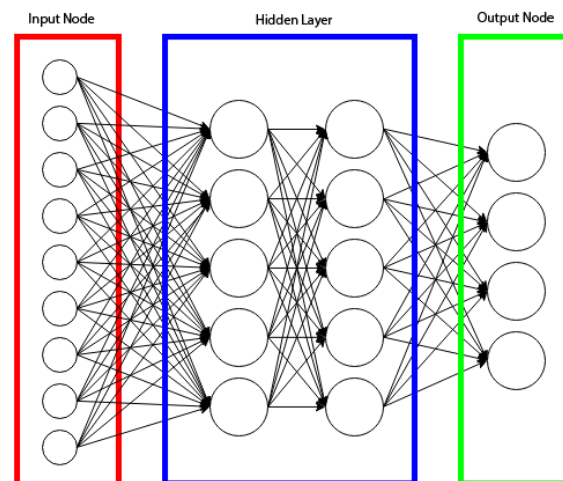


Figure 1. Network Architecture

2.3. Naive Bayesian

The naïve Bayesian method are the one of the simplest method on machine learning. This method calculate the probability on every single data for every single class using statistical approaches. Below are the formula to calculate the probability.

$$\bullet \quad P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

Where $P(c|x)$ is the posterior probability of class (target) given predictor (attribute), $P(c)$ is the prior probability of class, $P(x|c)$ is the likelihood which is the probability of predictor given class, $P(x)$ is the prior probability of predictor.

3. Methodology

In this section, the step to conduct the experiment for this paper will be discussed. This includes the data, how the pre-processing was done, how the network was built, and the parameter this paper used. The workflow of this paper can be seen on Figure 2.

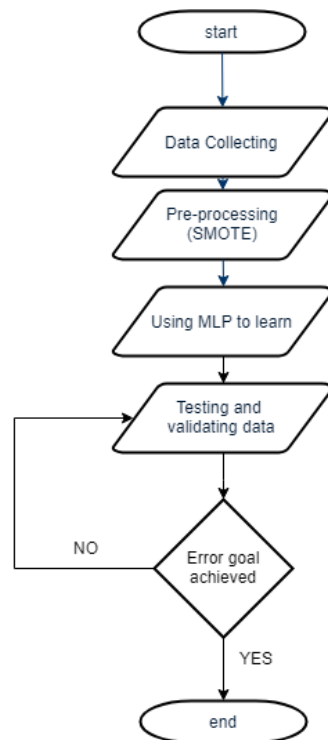


Figure 2. Workflow of this paper

3.1. The Data

This paper used the data from [4] which content is the leaving bed and chair motion for healthy elderly people. This dataset have 8 features and for each data. We used about 22.000 data from this dataset. The data are a sequential data where the data are taken every set of times.

3.1.1. Pre-Processing The Data

The data however, have an imbalance amount of data for each class, so in order to balance the class, we used SMOTE method. To calculate the input for parameter on the method by this formula :

$$F = \frac{M}{N} * 100\% - 100\%$$

Where F are the input for the parameter, M are the number of data on majority class, and N are the number of data on the minority class we want to oversample, we subtract the data with 100% because this 100% are the initial value of the minority data, if we don't subtract the data, then the minority data number will ended more than the majority data, causing more imbalance data.

3.2. Network Architecture

After pre-processing the data, next is building a model to classify the data. Before validating the data, this paper tweak the parameter on the model, the main focus here is the increase of the hidden layer and the learning rate. As for the neurons, 5 neurons was used for each hidden layer. The experiment was done multiple times to find the best result for each parameters. First, to find the best number of hidden layer, a static learning rate was used, which is 0.3. After finding the best number of hidden layer, next is to find the best number of learning rate. Like before, this was done by using a static hidden layer, used from the result before.

3.3. Validating The Data

The model are evaluated by using the cross-validation method using 10 folds. In order to measure the result of the fold, some function was used, named mean squared error (MSE) and mean average error (MAE) which formula can be seen below.

- $MSE = \frac{1}{N} \sum_{i=1}^x (A_{ij} - E_{ij})^2$
- $MAE = \frac{1}{N} \sum_{i=1}^x (A_{ij} - E_{ij})$

The functions above shows how to calculate mean squared error (MSE) which is the mean of squared differences between the actual value and the predicted value, mean average error (MAE) which is the average of all the errors in a set, A means the actual value from the dataset and E means predicted value of the data.

4. Result and Discussion

In this section, the result of this experiment will be discussed. The result of this experiment could be seen on Table 1 where H means the number of hidden layer and L means the learning rate.

Table 1. Experiment Result

	Accuracy	MSE	MAE
H=1 L=0.3	0.8953	0.1964	0.0725
H=2 L=0.3	0.9017	0.1896	0.0704
H=3 L=0.3	0.8994	0.1912	0.0697
H=4 L=0.3	0.8968	0.1963	0.0731
H=2 L=0.4	0.8941	0.1957	0.0734
H=2 L=0.5	0.8914	0.1968	0.0726

The accuracy of increased hidden layer can be seen on Figure 3. The performance of the model started dropping when the hidden layer are greater or equal 3.

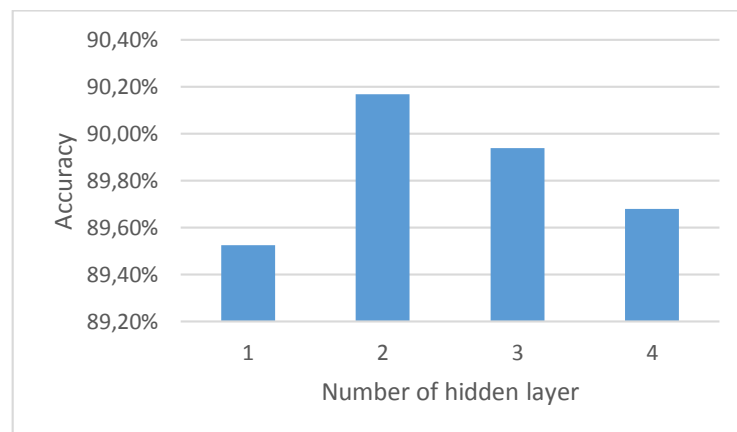


Figure 3. Accuracy on Increased Hidden Layer

The accuracy of increased learning rate can be seen on Figure 4. The more learning rate used, the less the accuracy we got.

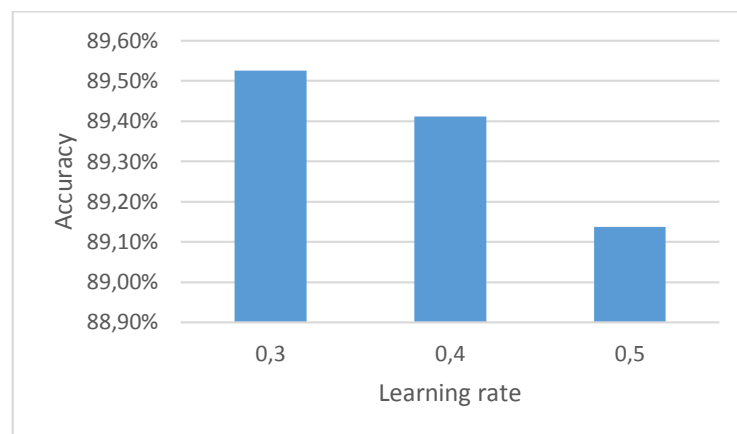


Figure 4. Accuracy on Increased Learning Rate

Naïve Bayesian method in other hands, only produces 84.46% accuracy. This is because there's no learning process in naïve Bayesian method, so there's no improvement on the learning process, only the first built model will be used, no improvement added. This is one of the reason MLP surpasses the naïve Bayesian method, because MLP will iterate through the data and improve the model to find the best possible weight value it can find.

5. Conclusion

After conducting all the experiment, we can conclude that increasing the number of hidden layer or learning rate will not increase the accuracy of the model we built. As why this can happened, the increase on hidden layer can make the network to learn unnecessary feature, so that the unnecessary feature will lead the recognition process to the wrong way. Thus, MLP can conduct a good result on the recognition of human motion on leaving the bed or chair. The naïve Bayesian method in other hand, produces a less accurate result compared to MLP.

6. References

- [1] Krishnan, N. C., & Cook, D. J. 2014. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*, 10, 138–154. <https://doi.org/10.1016/j.pmcj.2012.07.003>.
- [2] Ronao, C. A., & Cho, S. 2016. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems With Applications*, 59, 235–244. <https://doi.org/10.1016/j.eswa.2016.04.032>.
- [3] Yassine, A., & Singh, S. 2017. SPECIAL SECTION ON ADVANCES OF MULTISENSORY SERVICES AND TECHNOLOGIES Mining Human Activity Patterns From Smart Home Big Data for Health Care Applications, 5, 13131–13141.
- [4] Torres, R. L. S., Ranasinghe, D. C., Shi, Q., & Sample, A. P. 2013. Sensor Enabled Wearable RFID Technology for Mitigating the Risk of Falls Near Beds, 191–198.
- [5] Hengel, A. Van Den, & Ranasinghe, D. C. (n.d.). Effectiveness of a Batteryless and Wireless Wearable Sensor System for Identifying Bed and Chair Exits in Healthy Older People. <https://doi.org/10.3390/s16040546>.
- [6] Attal, F., Mohammed, S., Dedabrishvili, M., & Chamroukhi, F. 2015. Physical Human Activity Recognition Using Wearable Sensors, 31314–31338. <https://doi.org/10.3390/s151229858>.
- [7] Cornacchia, M., Member, S., Ozcan, K., & Member, S. 2016. A Survey on Activity Detection and Classification using Wearable Sensors, (NOVEMBER), 1–17. <https://doi.org/10.1109/JSEN.2016.2628346>.
- [8] Jaiswal, D., & Gigie, A. 2018. Table of Interest : Activity Recognition and Behaviour Analysis Using a Battery Less Wearable Sensor, 39–44.
- [9] Semwal, V. B., Raj, M., & Nandi, G. C. 2014. Multilayer Perceptron Based Biometric GAIT

Identification. *Robotics and Autonomous Systems*.
<https://doi.org/10.1016/j.robot.2014.11.010>.

- [10] Pham, B. T., Bui, D. T., Pourghasemi, H. R., & Indra, P. 2015. Landslide susceptibility assessment in the Uttarakhand area (India) using GIS : a comparison study of prediction capability of naïve bayes , multilayer perceptron neural networks , and functional trees methods. <https://doi.org/10.1007/s00704-015-1702-9>.
- [11] Hemmat, M., Afrand, M., Wongwises, S., Naderi, A., Asadi, A., Rostami, S., & Akbari, M. 2015. Applications of feedforward multilayer perceptron artificial neural networks and empirical correlation for prediction of thermal conductivity of Mg (OH) 2 – EG using experimental data ☆. *International Communications in Heat and Mass Transfer*, 67, 46–50. <https://doi.org/10.1016/j.icheatmasstransfer.2015.06.015>.
- [12] Jiang, L., Li, C., Wang, S., & Zhang, L. 2016. Engineering Applications of Artificial Intelligence Deep feature weighting for naive Bayes and its application to text classification. *Engineering Applications of Artificial Intelligence*, 52, 26–39. <https://doi.org/10.1016/j.engappai.2016.02.002>.
- [13] Chawla, N. V, Bowyer, K. W., & Hall, L. O. 2002. SMOTE : Synthetic Minority Over-sampling Technique, 16, 321–357.

Acknowledgements

This article is presented at the International Conference on Smart City Innovation 2018 that supported by the United States Agency for International Development (USAID) through the Sustainable Higher Education Research Alliance (SHERA) Program for Universitas Indonesia's Scientific Modeling, Application, Research and Training for City-centered Innovation and Technology (SMART CITY) Project, Grant #AID-497-A-1600004, Sub Grant #IIE-00000078-UI-1.