Journal of Korea Multimedia Society Vol. 23, No. 3, March 2020(pp. 447-459)
https://doi.org/10.9717/kmms.2020.23.3.447

IoT Device Classification According to Context-aware Using Multi-classification Model

Xu Zhang[†], Shinhye Ryu[††], Sangwook Kim[†††]

ABSTRACT

The Internet of Things(IoT) paradigm is flourishing strenuously for the last two decades. Researchers around the globe have their dreams to transmute every real-world object to the virtual object. Consequently, IoT devices are escalating exponentially. The abrupt evolution of these IoT devices has caused a major challenge i.e. object classification. In order to classify devices comprehensively and accurately, this paper proposes a context-aware based multi-classification model for devices, which classifies the smart devices according to people's contexts. However, the classification features of contextual data of different contexts are difficult to extract. The deep learning algorithm has the capability to solve this problem. This paper proposes a context-aware based multi-classification model of devices, which classifies the smart devices according to people's contexts.

Key words: Multi-classification, Internet of Things, Context-aware, Deep Learning

1. INTRODUCTION

By means of external data, human activity recognition (HAR) is capable of identifying and analyzing action types and behavior patterns. It is the main method for the computer to detect and understand people actions and behaviors [1]. The strengthening of sensor hardware makes automatic supervision and monitoring of daily behaviors possible [2]. Because of IoT sensors and devices can be applied to various scenarios, it is possible to collect data from devices and analyze reliability and availability of these data. So as to provide reliable context information for user-centered services [3]. With the rise of the artificial intelligence theory, as the field of application continues to expand, new challenges have emerged in terms of classification issues [4].

First, most practical problems can be deemed as multi-class problems, while the traditional method is usually designed for binary classification problems, and thus it is necessary to extend binary classification to multiple classifications [5-6]. By describing existing classes, it is possible to find a set of novel samples dense enough to be used as training samples for new classes [7]. Another idea is to consider all categories simultaneously in an optimization problem, which is called "all-together" method [8-9]. However, this kind of method has a long computing time and its performance is not ideal when dealing with large-scale data.

In order to classify the devices according to the

※ Corresponding Author : Sangwook Kim, Address: (702-701) IT No.4-401, Kyungpook National University, Daehakno 80, Bukgu, Daegu, Korea, TEL : +82-53-940-8881, E-mail : kimsw@knu.ac.kr
Receipt date : Aug. 12, 2019, Revision date : Jan. 29, 2020
Approval date : Feb. 5, 2020
[†] School of Computer Science and Engineering, Kyungpook National University
(E-mail : z.xu1993@gmail.com)
[††] School of Computer Science and Engineering, Kyungpook National University
(E-mail : shryu@media.knu.ac.kr)
[†††] School of Computer Science and Engineering, Kyungpook National University
※ This research was supported by the BK21 Plus project of the Ministry of Education and the Korea Research Foundation (SW Human Resources Development Team for the realization of Smart Life at Kyungpook National University) (21A20131600005)

real-time data of devices, this paper trains and calculates the historical data of the devices under different contexts. According to people's current context, the system can provide customized services for people. For example, if the system judges that the people's current context is getting up, it automatically provides a series of wake-up services for people, such as turning on lights and alerting clock. If the system judges that the people's current context is breakfast preparation, it automatically sends healthy food information and other considerations to people. If the system judges that the people's current context is going back to sleep, it automatically prepares relevant sleeping materials for people. With device classification information, the system captures people's behavior and provides people with more convenient service.

This paper consists of the following sections. Section I, *Introduction*, introduces the research background and its significance. Section II, *Related Works*, describes related multi-classification methods. Section III, *Solution of Multi-classification,* introduces the algorithms and formulas of proposed multi-classification model. Section IV, *Experimental Results and Analysis,* is record of experimental result and accuracy comparison of different model parameter. Section V, *Conclusions*, presents the optimal parameter of proposed multi-classification model and its advantage and disadvantage.

2. RELATED WORKS

For the multi-classification problem of devices, Zhang et. al. [10] uses a method called Machine Learning Naive Bayes (MLNB). This method adapts the traditional Naive Bayes classifier to handle multi-label classification problems. Although MLNB-basic enables Naive Bayes to learn multi-label instances, there are two factors negatively impact its performance. First, MLNB-basic assumes categories to be independent of each other with classic Naive Bayes. However, this assumption is usually not used and may reduce learning performance. Second, MLNB-basic solves the multi-classification problem by breaking down the multi-label learning problem into a number of independent binary learning problems (one for each tag). In this way, the correlation between the different tags of each instance is carelessly ignored and the performance of the algorithm may be penalized. In addition, related research is based on the purpose of multi-classification by the K-Nearest Neighbor (KNN) algorithm. KNN is a supervised learning algorithm, in which the results of new queries are classified based on most K-Nearest Neighbor categories [11]. The KNN classification algorithm is used to support online training and classification, which reduces the classification execution time, but the classification accuracy is not satisfactory [12].

In addition, incremental learning algorithm ID4 realizes incremental learning of decision tree, however it simply abandons the subtree whose segmentation attribute changes and wastes historical calculation. Besides, small number of positive samples and a large number of unlabeled samples are used for multi-classification learning, and several real-world applications require real-time prediction on resource-scarce devices, such as IoT sensors [13-15]. All kinds of multi-classification learning methods have semi-supervised learning ability [16-17]. The constraints contained in labeled data are introduced to implement semi-supervised clustering [18].

In order to conveniently manage the devices of IoT, the advantage of semantic method in context organization is pointed out, but accuracy is not ideal [19]. Context recognition and cross-validation are implemented by introducing a set of multi-classification methods with extracted features. Besides, with the help of data from multiple sensors, performance is improved. However, due to the large number of outliers generated in the analysis device data, the classification effect is decreased [20]. In order to essentially solve the problems mentioned above, a deep learning multi-classification model which is based on context-aware has constructed. Specifically, the deep learning algorithm of this paper is used to improve the classification accuracy of IoT devices.

3. SOLUTION OF DEVICE MULTI-CLASSIFICATION

In order to accurately and comprehensively perceive people's context, the dataset introduced in this paper collects sensor data of various smart devices. For example, smart devices installed on the body and environment object smart devices in people life context. In addition, the categories of devices are based on combination of body actions and environment objects operated by people, such as opening the door, closing the door, opening the fridge, closing the fridge, and so on. With the help of this dataset, the deep learning algorithm is used to train and learn multi-classification model. Finally the context-aware multi-classification model of IoT device is obtained. This section introduces the implementation of the solution and process.

3.1 Data Acquisition of Device Multi–classification

In order to achieve the purpose of classifying devices in different contexts, this paper adopts the dataset which utilizes multiple sensors (such as accelerations, gyroscopes, and switches) deployed in environment and body. In Table 1, this dataset contains a total of 13 actions, 23 environment objects and 17 activity categories.

3.2 The Overview of Device Multi–classification

In Fig. 1, the multi-classification model consists of input layer, data pre-treatment layer, hidden layer and output layer.

In data pre-treatment layer, since the influences of all attributes are independent, therefore, the softmax function is used to conduct data pre-treatment process. The main operation of the softmax function is to add all the data and calculate the percentage of a value to the total. At the same time, due to the sensor hardware problems and acquisition environment problems, the data is inevitably empty or offset. Therefore, the data need to be logically repaired.

Table 1. Different Characteristics of the Dataset

Action	Object	Classification
Unlock	Bottle	Drink from cup
Stir	Salami	Open fridge
Lock	Bread	Close fridge
Close	Sugar	Clean table
Reach	Dishwasher	Open door 1
Open	Switch	Open door 2
Sip	Milk	Close door 2
Clean	Drawer 3	Close door 1
Bite	Spoon	Open dishwasher
Cut	Knife cheese	Toggle switch
Spread	Drawer2	Close dishwasher
Release	Table	Open drawer 3
Move	Glass	Close drawer 3
	Cheese	Open drawer 1
	Chair	Close drawer 1
	Door 1	Open drawer 2
	Door 2	Close drawer 2
	Plate	
	Drawer 1	
	Fridge	
	Cup	
	Knife salami	
	Lazy chair	

In hidden layer, since the limitation of linear logic relation, nonlinear conversion method is used to transform data non-linearly. Nonlinear conversion is performed on input data. Combining different nodes in hidden layer with weight parameter of different input data attributes, the node's output value is calculated.

In output layer, in order to attain the output values of hidden layers, the calculations has been performed layer by layer. In this paper, this model can

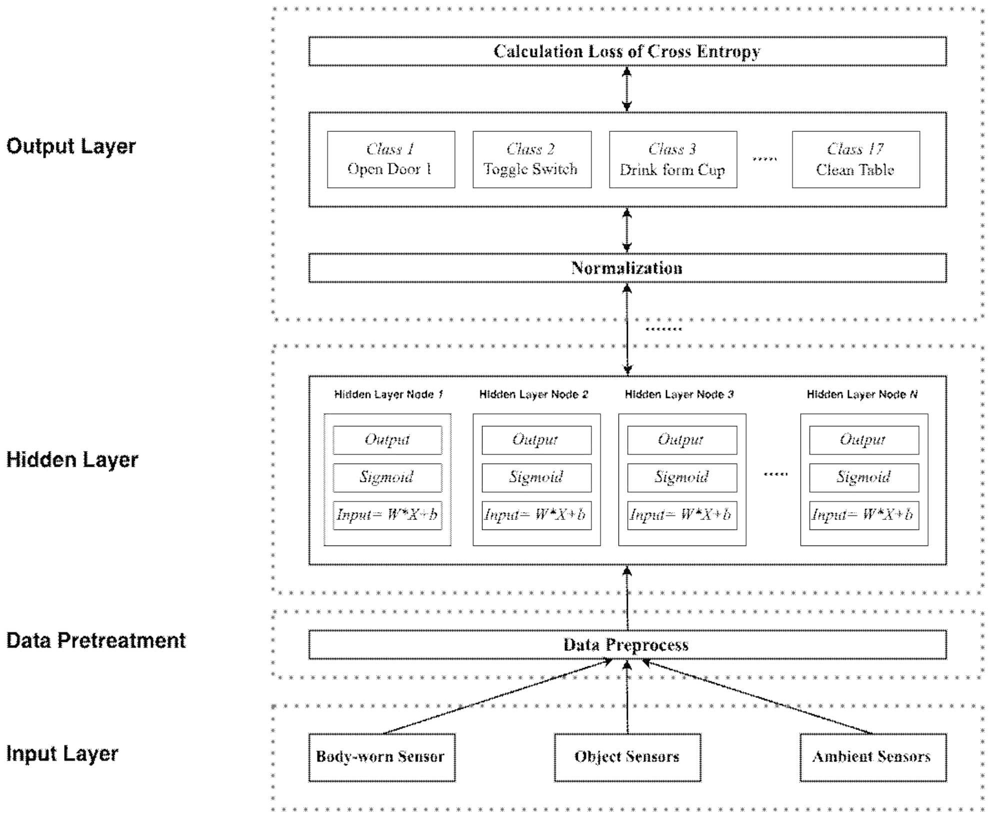


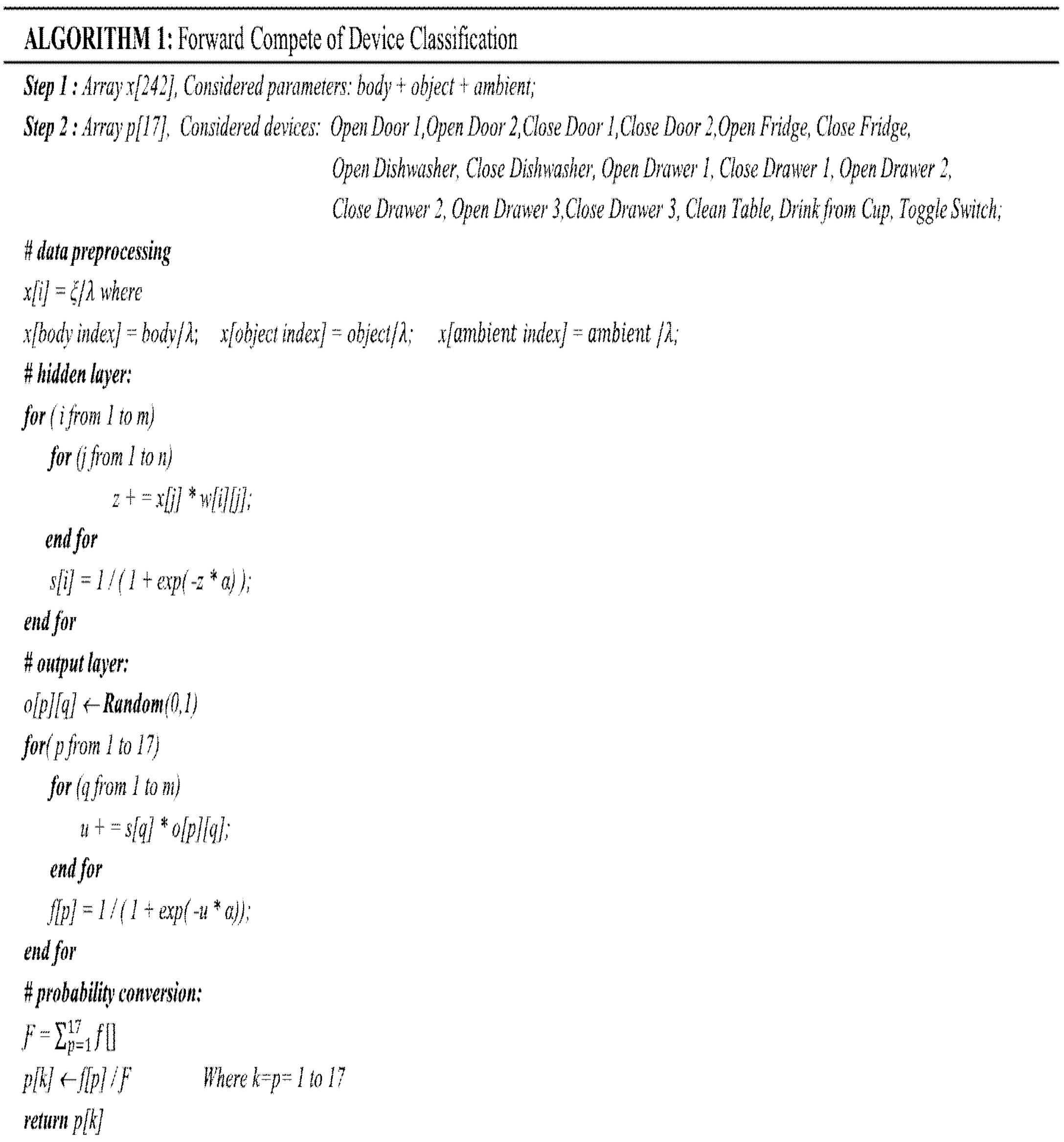
Fig. 1. Structure of Device Classification for Different Contexts.

classify 17 types, so there are 17 nodes in the output layer. Since the values of these nodes do not meet the probability distribution, the softmax function carries out probability conversation on the values of output layer, and calculates the probability value of result. These probability values add up to 1, meaning that any set of input data must be part of the 17 categories.

3.3 The Algorithms of Device Multi-classification

The calculation process of the multi-classification model includes feedforward calculation and reverse optimization calculation. Algorithm 1 preprocesses and non-linear processes the data collected by the devices, then calculates the value of output layer, and finally converts it into the probability relationship with the corresponding classification result. The probability result and the label value of the input data are compared by algorithm 2 to calculate the loss value. Finally, the gradient weighting function is used to optimize the weight parameter of the algorithm, so as to obtain the final multi-classification model. The details of the implementation are shown in the follow figures.

In Fig. 2, m is the node number of hidden layer; n is the attribute number of input layer; $w[][]$ is the weight parameter of node of hidden layer; z is the summary value of node of input layer; α is the sigmoid function parameter; $s[]$ is the output value array of node of hidden layer; $o[][]$ is the weight parameter of node of output layer; u is the summary value of node of hidden layer; $f[]$ is the output value array of output layer; $p[]$ represents the probability values of device type in turn. The sum of the values of the array p is 1.



ALGORITHM 1: Forward Compete of Device Classification

```
Step 1 : Array x[242], Considered parameters: body + object + ambient;
Step 2 : Array p[17], Considered devices: Open Door 1,Open Door 2,Close Door 1,Close Door 2,Open Fridge, Close Fridge,
                                          Open Dishwasher, Close Dishwasher, Open Drawer 1, Close Drawer 1, Open Drawer 2,
                                          Close Drawer 2, Open Drawer 3,Close Drawer 3, Clean Table, Drink from Cup, Toggle Switch;
# data preprocessing
x[i] = $\xi/\lambda$ where
x[body index] = body/$\lambda$;   x[object index] = object/$\lambda$;   x[ambient index] = ambient /$\lambda$;
# hidden layer:
for ( i from 1 to m)
   for (j from 1 to n)
         z + = x[j] * w[i][j];
   end for
   s[i] = 1 / ( 1 + exp( -z * a) );
end for
# output layer:
o[p][q] ← Random(0,1)
for( p from 1 to 17)
   for (q from 1 to m)
        u + = s[q] * o[p][q];
   end for
   f[p] = 1 / ( 1 + exp( -u * a));
end for
# probability conversion:
$F = \sum_{p=1}^{17} f[]$
p[k] ← f[p] / F          Where k=p= 1 to 17
return p[k]
```

Fig. 2. Forward Compete of Device Classification.

In Fig. 3, m is the node number of hidden layer; n is the attribute number of input layer; $w[][]$ is the weight parameter of node of hidden layer; α is the sigmoid function parameter; $s[]$ is the output value array of node of hidden layer; $o[][]$ is the weight parameter of node of output layer; $e[]$ is the loss value between probability array $p[]$ and label data array $y[]$; β is the sigmoid function parameter to correct the effect of α; $h[]$ is the loss value array of hidden layer; $neweight$ is the correction steps size of output layer; $newstep$ is the correction steps size of hidden layer; $rate$ is the correction speed.

ALGORITHM 2: Back Propagation Compete of Device Classification

```
Step 1 : Array p[k], probability of 17 devices: Open Door 1,Open Door 2,Close Door 1,Close Door 2,Open Fridge,
                                                Close Fridge, Open Dishwasher, Close Dishwasher, Open Drawer 1,
                                                Close Drawer 1, Open Drawer 2, Close Drawer 2, Open Drawer 3,
                                                Close Drawer 3, Clean Table, Drink from Cup, Toggle Switch;
      Array y[j], include 17 devices: the same to array p[k];
                 Where        j=k= 1 to 17
Step 2 : Multi classification model;
# data preprocessing:
y[1]  ←  [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
y[2]  ←  [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]  ...
y[17] ←  [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]
# output layer:
for (i from 1 to 17)
    e[i] ← ( y[i] - p[i] ) * p[i] * ( 1 - p[i] ) * α * β;
end for
for ( i from 1 to 17)
    for (j from 1 to m)
        newstep ← rate * e[i] * s[j];
        o[i][j] ← o[i][j] + newstep;
    end for
end for
#hidden layer
for (i from 1 to m)
    for (j from 1 to 17)
        sum + = o[j][i] * e[j];
    end for
    h[i] ← sum * s[i] * ( 1 - s[i]) * α * β;
end for
for ( i from 1 to m)
    for ( j from 1 to n)
        neweight ← rate * h[i] * x[j];
        w[i][j] ← w[i][j] + neweight;
    end for
end for
return multi classification model
```

Fig. 3. Back Propagation Compete of Device Classification.

3.4 The Formulas of Device Multi-classification

This section details some of the important formulas involved in the multi-classification model. Due to the difference of the devices and the influence of the data collection environment, there is a inevitable error in the dataset, so the data needs to be pre-processed. As in formula (1):

$$f(x) = \frac{x - \frac{1}{n}(\sum_{i=0}^{n} x_i)}{|x_{\max}|} \tag{1}$$

The input layer has n attributes, and i represents any of them. Data collected by devices are assumed to be $x^1_{attribute}, x^2_{attribute}, x^3_{attribute}, \cdots, x^n_{attribute}$. These data are filtered, reorganized and converted as dataset of this multi-classification model. The input value of one neural nodes can be calculated by using formula (2):

$$S_j = \sum_{i=0}^{n} w^{ij}_{weight} x^{i}_{attribute} + b_j \quad (0 < j < m) \tag{2}$$

In formula (2), m is the number of nodes of hidden layer, and S_j is the input value of j-th node of one layer, w^{ij}_{weight} is the weight parameter of the j-th node of the current layer to the i-th node of the previous layer, $x^{i}_{attribute}$ is the output value of the i-th node of the previous layer or the attribute value of the i-th node of the input layer, b_j is the constant parameter used to make up the deviation. In order to avoid the input value entering to follow sigmoid activation function's saturation interval, the optimization parameter $\alpha\,(0<\alpha<1)$ is set, and then the input value of the corresponding node is calculated by the formula (3):

$$S_j = \alpha (W^{j}_{weight})^T X_{attribute} + b_j \quad (0<j<m) \tag{3}$$

In formula (4), sigmoid activation function is used to change the input value to nonlinear data. In formula (4), $Output(s_j)$ is the output value of j-th node of one layer.

$$Output\,(s_j) = \frac{1}{1+e^{-s_j}} \quad (0<j<m) \tag{4}$$

Since the output value of last layer of the multi-classification model doesn't meet the probability distribution, the softmax function needs to be used for normalization. In formula (5), k is the node number of output layer, and P_t is the probability value of the t-th node in the output layer.

$$P_t = \frac{Output_{s_t}}{\sum_{t=1}^{k} Output_{s_t}} \quad (0<t<k) \tag{5}$$

After that, the cross-entropy C is used to calculate the loss value, which can represent the accuracy between the truth value and the output value. y is the label value of training sample value of input layer. Because each training time in this paper inputs H samples. So the loss function C of the algorithm for batch processing H samples is:

$$C = -\frac{1}{H}\sum_{h}^{H}\left[y\, Log\, Ouput_s + (1-y)\, Log\,(1-Output_s)\right] \quad (0<h<H) \tag{6}$$

Next, loss value should be back-propagated to each node of the multi-classification model to adjust the previous node weight parameter, so as to obtain higher accuracy results in the next calculation. Finally, from formula (2) to formula (6) need to be cycled many times according to different input data. In order to control the speed of optimizing weight parameter, the learning $rate$ is set, through which the optimal rhythm of weight can be reasonably controlled.

$$C'(W_{weight}) = \nabla = \frac{C(W^{n+1}_{weight}) - C(W^{n}_{weight})}{W^{n+1}_{weight} - W^{n}_{weight}} \tag{7}$$

$$W^{n+1}_{weight} = W^{n}_{weight} - rate\nabla = W^{n}_{weight} - rate\, C'(W_{weight}) \tag{8}$$

Finally, by the formula (7) (8), optimal weight parameter is backwardly calculated layer by layer, and the optimal weight parameter is obtained. With more and more input sample data, the classification result is more and more accurate. Finally, a reasonable classification model is obtained.

4. EXPERIMENTAL RESULTS AND ANALYSIS

This section introduces the result of multi-classification model. With the help of the dataset "*Wearable Computing Laboratory, ETH Zürich, Switzerland dataset*" [21], during in the training process of the deep learning algorithm, the same algorithm parameters are set. The accuracy of different classifications is shown in Table 2.

In addition, setting different parameters for the deep learning algorithm gets different classification accuracy. Experiment result shows that the number of hidden layers has a certain impact on the accuracy of the classification result.

In Fig. 4, for "Open dishwasher", when the number of hidden layer is 2, the accuracy is higher. For "Open drawer 2", when the number of hidden layer is 3, the accuracy is higher.

Similarly, the number of hidden layer nodes in each layer of the multi-classification model

also has an impact on the accuracy of the classification result.

Table 2. Accuracy of Classification under the Same Parameters

Classification	Sample Size (Units: Number)	Feature Dimension	Learning Rate	Precision Rate (%)
Drink from cup	13800	242	0.01	98%
Open fridge	13800	242	0.01	91%
Close fridge	13800	242	0.01	88%
Clean table	13800	242	0.01	97%
Open door 1	13800	242	0.01	94%
Open door 2	13800	242	0.01	97%
Close door 2	13800	242	0.01	98%
Close door 1	13800	242	0.01	97%
Open dishwasher	13800	242	0.01	98%
Toggle switch	13800	242	0.01	98%
Close dishwasher	13800	242	0.01	96%
Open drawer 3	13800	242	0.01	96%
Close drawer 3	13800	242	0.01	82%
Open drawer 1	13800	242	0.01	89%
Close drawer 1	13800	242	0.01	94%
Open drawer 2	13800	242	0.01	85%
Close drawer 2	13800	242	0.01	96%

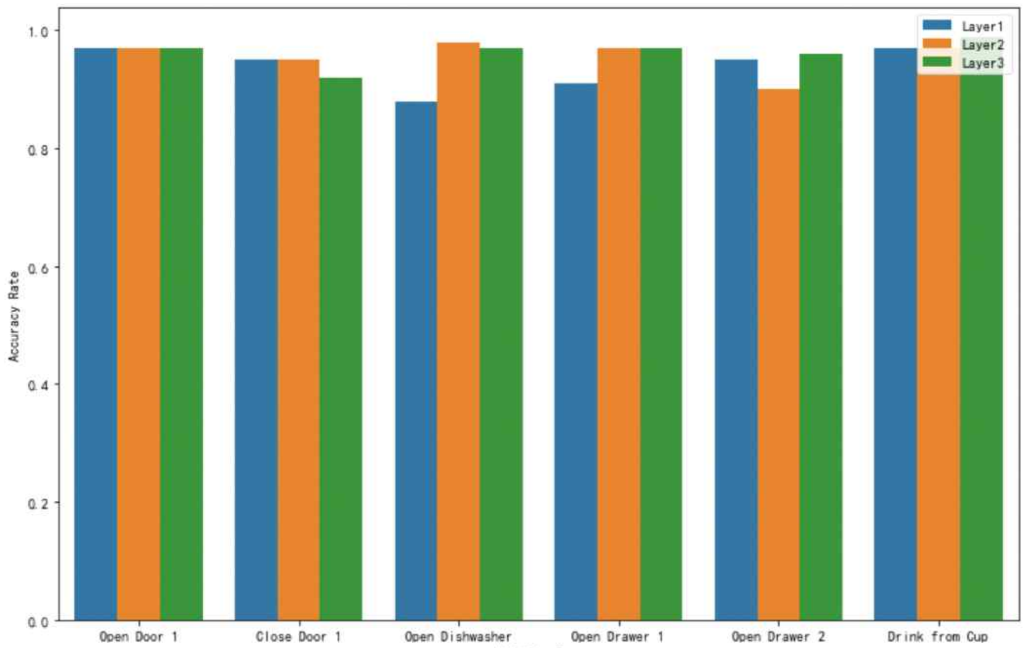


Fig. 4. Accuracy of Different Layers.

In Fig. 5, for "Open door 1", when the number of nodes in the hidden layer is 512*512 (the number of nodes in the first hidden layer is 512, and the number of nodes in the second hidden layer is 512), the accuracy is higher. For "Open drawer 2", the accuracy is higher when the number of hidden layer nodes is 1024*1024.

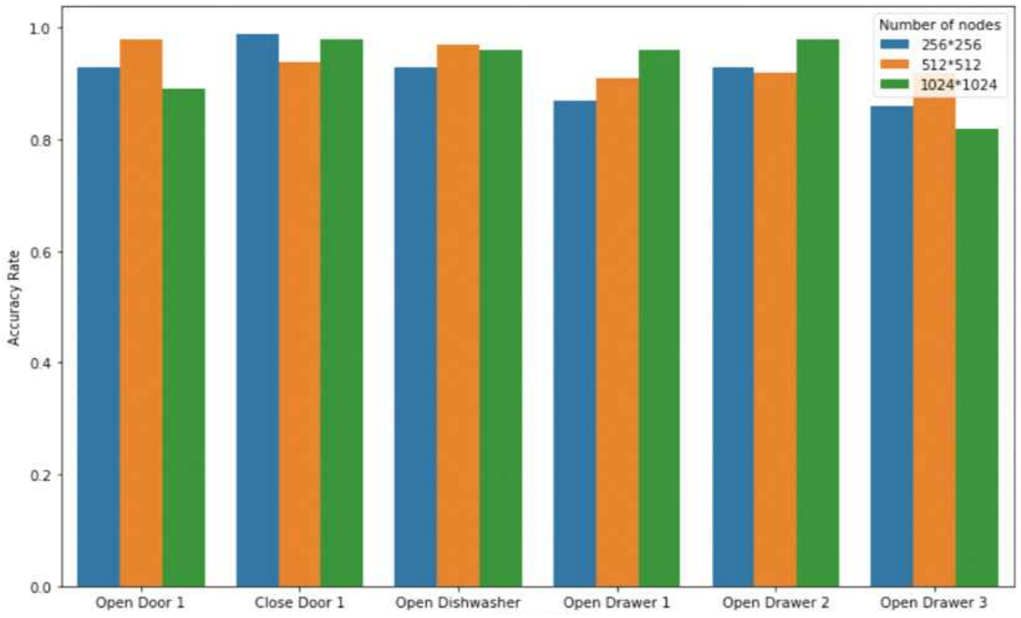


Fig. 5. Accuracy of Different Nodes.

Besides, in the deep learning algorithm, different learning rates get different classification accuracy values. In Fig. 6, this paper classifies 17 types of devices and sets the learning rate in the $(10^{-6}, 10)$ interval to find the optimal learning rate. Experimental result shows that the accuracy of the classifier is higher when the learning rate is 0.01. However, the learning rate cannot be set too large or too small. When the learning rate is too large, the accuracy is low. This is because when the learning rate is set to a small value, the convergence speed of the loss function becomes small, and it is very easy to converge at a local minimum value. When the learning rate is set too large, the loss function easily misses the optimal value and fluctuates around the optimal value. So setting a appropriate learning rate value is very important to get the optimal accuracy value.

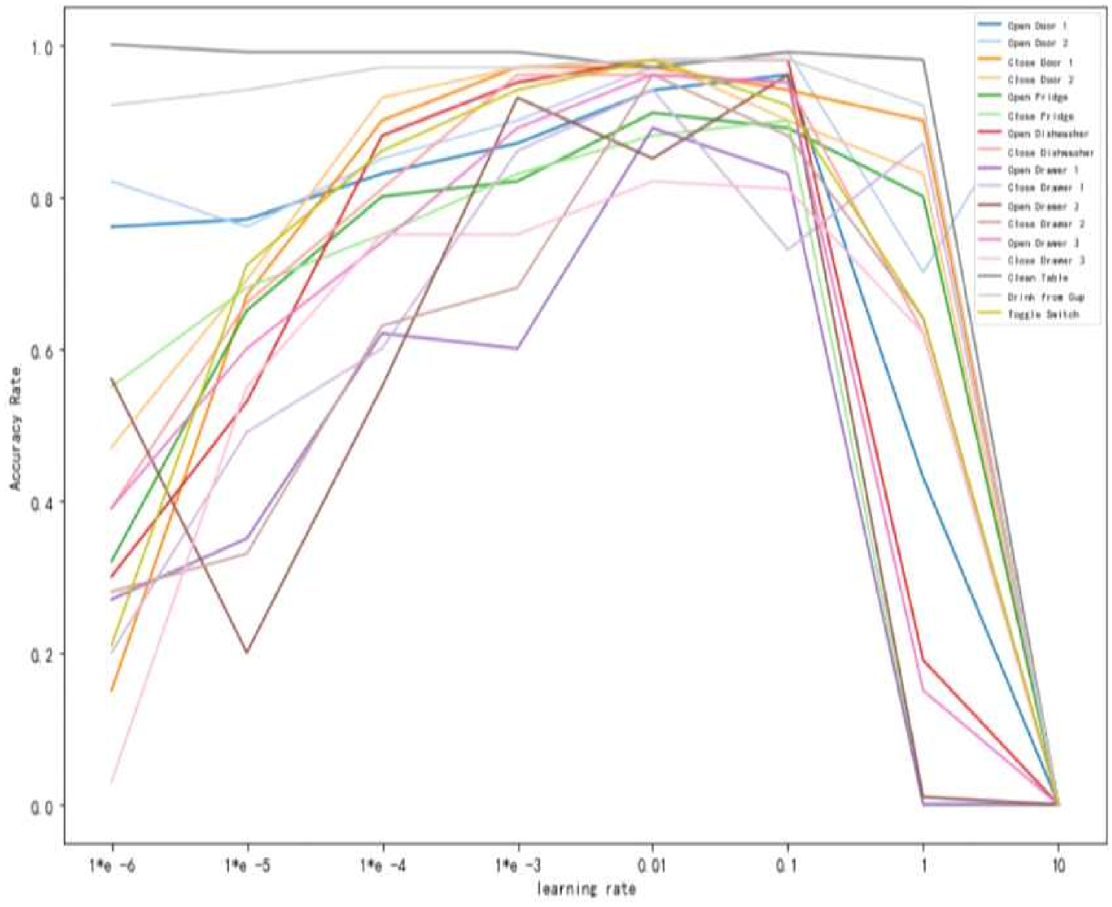


Fig. 6. Accuracy Rate for Different Learning Rates.

In Table 3, the dataset of this paper is imported into Bayesian algorithm and KNN algorithm, and the accuracy of 17 classification results is counted. It can be seen from the results, compared with NB and KNN algorithms, the accuracy of context- aware deep learning algorithm proposed in this paper is much more higher.

Table 3. Accuracy of Classification for Different Methods base on the Same Dataset

Multi-classification Methods	Sample Size (Units: Number)	Feature Dimension	Average Accuracy(%)	
Naive Beyesian (*NB*)	234600	242	Open door 1	53.33%
			Open door 2	71.04%
			Close door 1	57.2%
			Close door 2	42.99%
			Open fridge	57.52%
			Close fridge	49.68%
			Open dishwasher	51.79%
			Close dishwasher	56.42%
			Open drawer 1	42.06%
			Close drawer 1	40.87%
			Open drawer 2	53.23%
			Close drawer 2	27.77%
			Open drawer 3	41.3%
			Close drawer 3	32.49%
			Clean table	81.40%
			Drink from cup	79.95%
			Toggle switch	37.63%
K-Nearest Neighbor (*KNN*)	234600	242	Open door 1	81.59%
			Open door 2	91.29%
			Close door 1	14.28%
			Close door 2	7.34%
			Open fridge	7.35%
			Close fridge	12.65%
			Open dishwasher	5.44%
			Close dishwasher	8.45%
			Open drawer 1	28.96%
			Close drawer 1	27.88%
			Open drawer 2	29.50%
			Close drawer 2	40.95%
			Open drawer 3	9.90%
			Close drawer 3	6.73%
			Clean table	38.09%
			Drink from cup	76.33%
			Toggle switch	8.92%
Context-aware Multi-classification (Proposed by this paper)	234600	242	Open door 1	97%
			Open door 2	97%
			Close door 1	96%
			Close door 2	97%
			Open fridge	93%
			Close fridge	89%
			Open dishwasher	97%
			Close dishwasher	98%
			Open drawer 1	77%
			Close drawer 1	99%
			Open drawer 2	92%
			Close drawer 2	89%
			Open drawer 3	87%
			Close drawer 3	97%
			Clean table	99%
			Drink from cup	98%
			Toggle switch	98%

5. CONCLUSIONS

The context-aware deep learning algorithm proposed in this paper can well classify devices. When the number of hidden layer is 2, the number of hidden layer nodes is 512, the learning rate is set to 0.01, and the dataset is trained 6 times, the average accuracy rate of 17 types can reach 95%. At the same time, the classifier can automatically extract data features from the context information, realize classification of devices, and save a lot of data collection workload.

To sum up, the multi-classification model in this paper has the following advantages. First: multi-classification of devices corresponding to varied context, such as location, activity, and other context of people. Second: the types of classification can be expanded easily. Third: reinforcement learning and optimization can escalate the accuracy.

The difficulties encountered at present are follows. First: high quality of training data leads to relatively high cost of data collection. Second: the optimal parameters of the context-aware deep learning algorithm need to be obtained through a large number of experiments.

In future research, it is necessary to reduce dependence on data volume and shorten training model time.

REFERENCE

[1] Y.F. Chen and C. Shen, "Performance Analysis of Smart Phone-sensor Behavior for Human Activity Recognition," *IEEE Access*, Vol. 5, pp. 3095-3110, 2017.

[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L.R. Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smart Phone," *Proceeding of European Symposium on Artificial Neural Networks*, pp. 437-442, 2013.

[3] S. Ryu and S. Kim, "Development of an Integrated IoT System for Searching Dependable Device based on User Property," *Journal of Korea Multimedia Society*, Vol. 20, No. 5, pp. 791-799, 2017.

[4] S.J. Russell and P. Norvig, *Artificial Intelligence- A Modern Approach*, Third Edition, Pearson Education, New Jersey, USA, 2010.

[5] C.W. Hsu and C.J. Lin, "A Comparison of Methods for Multi Class Support Vector Machines," *IEEE Transaction on Neural Networks*, pp. 415-425, 2012.

[6] E.J. Bredensteiner and K.P. Bennett, "Multi Category Classification by Support Vector Machines," *Computational Optimization and Applications*, 12(1/3), pp. 53-79, 1999.

[7] D.M.J. Tax, "One-class Classification, Concept Learning in the Absence of Counter Example," *PhD Thesis*, *Delft University of Technology*, 2001, http://www-ict.et.tudelft.nl/~davidt/papers/thesis.pdf, accessed 3 Sep 2009.

[8] V.N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[9] K. Crammer and Y. Singer, "On the Learnability and Design of Output Codes for Multiclass Problems," *Machine Learning*, Vol. 47, No. 2-3, pp. 201-233, 2002.

[10] M.L. Zhang, J.M. Peña, and V. Robles, "Feature Selection for Multi-label Naive Bayes Classification," *Information Sciences*, Vol. 179, No. 19, pp. 3218-3229, 2009.

[11] S. Kaghyan and H. Sarukhanyan, "Activity Recognition Using K-nearest Neighbor Algorithm on Smart Phone with Tri-axial Accelerometer," *International Journal of Informatics Models and Analysis,* Vol. 1, No. 2, pp. 146-156, 2012.

[12] P. Paul and T. George, "An Effective Approach for Human Activity Recognition on Smart Phone," *Proceeding of 2015 IEEE International Conference on Engineering and Technology*, pp. 1-3, 2015.

[13] K.P. Nigam, *Using Unlabeled Data to Improve Text Classification*, Ph.D's Thesis of Carnegie-Mellon University of Computer Science, 2001.

[14] C. Gupta, A.S. Suggala, A. Goyal, H.V. Simhadri, B. Paranjape, A. Kumar, et al., "ProtoNN: Compressed and Accurate KNN for Resource-scarce Devices," *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70, pp. 1331-1340, 2017.

[15] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning*, pp. 103-134, 2000.

[16] D. Bargeron, P. Viola, and P. Simard, "Boosting-based Transductive Learning for Text Detection," *Proceeding of Eighth International Conference on Document Analysis and Recognition*, pp. 1166-1171, 2005.

[17] F.G. Cozman, I. Cohen, and M.C. Cirelo, "Semi-supervised Learning of Mixture Models," *Proceedings of the 20th International Conference on Machine Learning*, pp. 99-106, 2003.

[18] M. Bilenko, S. Basu, and J.R. Mooney, "Integrating Constraints and Metric Learning in Semi-supervised Clustering," *Proceedings of the Twenty-first International Conference on Machine Learning*, pp. 81-88, 2004.

[19] M. Antunes, D. Gomes, and R.L. Aguiar, "Towards IoT Data Classification Through Semantic Features," *Future Generation Computer Systems*, Vol. 86, pp. 792-798, 2018.

[20] M.F.M. Fudzee and J. Abawajy, "A Classification for Content Adaptation System," *Proceedings of the 10th International Conference on Information Integration and Web-based Applications and Services*, pp. 426-429, 2008.

[21] R. Chavarriaga, H. Sagha, A. Calatroni, S.T. Digumarti, G. Tröster, D. Roggen, et al., "The Opportunity Challenge: A Benchmark Database for On-body Sensor-based Activity Recognition," *Pattern Recognition Letters*, Vol. 34, No. 15, pp. 2033-2042, 2013.

Xu Zhang

She recevied B.S. degree from Statistics Department of Kyungpook National University, Korea, in 2017. She received M.S. degree from Computer Science and Engineering of Kyungpook National University, Korea, in 2020. She is interested in mobile network, big data, AI, Internet of Things.

Shinhye Ryu

She received B.S. degree from Kyungpook National University, Korea, in 2016. She received M.S. degree from Computer Science and Engineering of Kyungpook National University, Korea, in 2018. She is currently completing her Ph.D in Computer Science and Engineering of Kyungpook National University. She is interested in mobile computing, social network, human computer Interaction. She is now a visiting professor at Kyungpook National University.

Sangwook Kim

He received B.S. degree from Kyungpook National University, Korea, in 1979. He received M.S. degree from Seoul National University, Korea, in 1981. He received Ph.D in Computer Science and Engineering from Seoul National University, Korea, in 1989. Since February 1982, he is Professor of School of Computer Science and Engineering at Kyungpook National University. He is interested in hyper-intelligence, social network computing, human computer interaction, mobile computing, digital media art, social media, Internet of Things, etc.