Multiclass Classification Method in Handheld Based Smartphone Gait Identification

Abdul Rafiez Abdul Raziff[1], Md Nasir Sulaiman[1], Norwati Mustapha[1], Thinagaran Perumal[1], Muhammad Syafiq Mohd Pozi[2]

[1]*Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*

[2]*Faculty of Computer Science and Engineering, Kyoto Sangyo University Motoyama, Kamigamo, Kita-ku, Kyoto-City 603-8555, Japan*

abdulrafiez@gmail.com

***Abstract*— Gait identification has been widely used in many types of research and application. Since gait identification involves with many people and classes, using a single classifier is not a good option as the dataset may contains overlapped class boundary and moreover, most of the classifiers are well built for binary classes. This paper discusses the application of multiclass classifiers such as one-vs-all (OvA), one-vs-one (OvO) and random correction code (RCC) on handheld based smartphone gait signal for person identification. The mapping uses J48 as the main classifier. The result is then compared with a single J48 for the benchmark. Finally, the best multiclass method is compared with few machine learning classifier in-order to see its capability. From the result, it can be seen that using OvO and RCC thus increase the accuracy performance if compared to a single classifier. For the best classifier in the multiclass mapping method, it can be seen that J48 yield the best accuracy score.**

***Index Terms*— Gait Identification; Multiclass Classification; OvA; OvO; RCC; Single Classifier.**

I. INTRODUCTION

Gait recognition using smartphone has been widely used in many research and applications such as biomechanics, neuro-rehabilitation, sport medicine, security, *etc.* [1]–[4] due to its lightweight, small in size, low power consumption and low development cost. Gait is a term that is referring to the term of a complete walking cycle which consist of a collection of steps. Gait identification is the process of identifying a person from the recorded gait signal.

Overlapped class boundary due to many classes in a dataset is an issue which makes a single classifier application is not efficiently enough [5]. Furthermore, according to Dietrich, most of the algorithms are best suited for binary classification functionality [6]. In this situation, multiclass classification mapping is an option to classify more than 2 classes. The rationale of using this method is to transform the multiclass dataset into smaller sets of binary classes dataset for an easier classification.

There are few multiclass methods that have the ability of classifying more than 2 classes (binary classes). Neural networks has the natural ability of classifying multiclass dataset situation. Few others existing machine learning algorithms also have the ability of classifying multiclass dataset such as decision trees, k-NN, SVM, and etc. Then, the extension of the multiclass classification method which is Extreme Learning Machine (ELM) [7] for single-hidden layer feedforward neural networks (SLFNs) that is able to choose randomly which hidden nodes and analytically determines the SLFNs' output weights. Unlike the traditional neural networks, the processing is slower due to the slow gradient-based learning and the parameters need to be adjusted iteratively.

The purpose of this paper is to investigate the accuracy performance of using multiclass classification mapping on handheld based signal for gait identification from smartphone. The methods used are one-vs-all (OvA), one-vs-one (OvO) and random correction code (RCC). These methods are then compared to a single J48 classifier for a benchmark purposes. Few machine learning classifiers algorithm are tested with the best multiclass classifier method in-order to find the most stable combination. The rationale of using multiclass classification is to increase the accuracy since the number of people using gait recognition can be many thus using a single classifier may not be a good choice to handle.

This paper is organized as follows; Section 2 discuss the related works. Section 3 describes the multiclass classification mapping. Section 4 discuss about other machine learning algorithm for evaluation. In Section 5, the experimental design is explained. Section 6 presents the experimental results and discussion and Section 7 presents the conclusion.

II. RELATED WORK

Single classification method was employed by [8]–[11] for gait recognition using an accelerometer. The result is quite convincing but it can be further improved in creating a more efficient and robust learning structure. However, when the single classifier is applied, it is difficult to predict the correct data point when the classes' boundary are overlapped in the hyper space.

In the code based multiclass classification, Exhaustive Correction Code (ECC) [6], the assignment of codes are based on the number of classes. If the number of class is between $3 \leq k \leq 7$, the code length would be $2^{k-1} - 1$. For each row, the number of 1 will be decreased by half. If the number of class is between $8 \leq k \leq 11$, the selection of the good subset of its column need to be perform after the construction of the exhaustive code. For a larger size of classes, using ECC may not be efficient enough as the size of the code length would reach thousands or millions which it is very expensive and impossible for computation.

Deep learning has become a platform for multiclass classification in named entity recognition in electronic medical records (EMRs) [12]. In this work, Convolutional

Neural Network (CNN) is applied for mining named entities from the EMRs. From the result, the applied model showed better score than NB and ME based model.

In another work, the author [13] conducted the experiment on the differential diagnosis on ADHD subtypes using Hierarchical Extreme Learning Machine (H-ELM) multiclass classification method from MRI images. From the result, H-ELM proved to be better than the standalone based classifier by 6%.

In the work conducted by [14], OvO multiclass classification mapping has been applied to the handheld based gait signal for people identification. Multilayer Perceptron (MLP), k-nearest neighbor (k-NN), Support Vector Machine (SVM) and J48 were been used as the based classifier. At the dataset level, SMOTE has been applied to increase the dataset size. From the result, MLP produced the best accuracy score followed by J48 and k-NN. OvO multiclass classification did outperform the single classifier mapping except on MLP which the overall accuracy are same with and without multiclass classification mapping.

In the wearable sensor experiment using smartphone, most of the researchers concentrate on different placement such as pockets, pouch, clipped on the clothes and other body parts [8]–[10], [15]–[18]. However, because of different clothing, culture and ability, hand held based smartphone placement is much more viable as it is more direct and fast (without hassle) when gait data need to be collected. However, based on the previous author's work, certain hand-held placement is not viable as it contains a high number of outliers.

III. MULTICLASS CLASSIFICATION MAPPING METHOD

In this section, four types of multiclass classification mapping methods are discussed. The process of binarization happens at the training dataset level before the training the model.

A. *Single J48 (Direct Classification)*

This method comprises of a single J48 [19] decision tree. It is a algorithm to generate decision tree based on ID3 algorithm [20]. The enhancement done on J48 is attribute that can compensate with continuous or discrete value. Besides that, it also uses a bottom-up method (pruning) to solve the over-fitting problem.

A single J48 is able to perform classification on multiclass problem as each leaf can be labelled to with one of the classes whilst the internal nodes can act to discriminate features among the classes [6].

B. *One-vs-one (OvO)*

One-vs-one (OvO) or another term is called as the pairwise classification is a multiclass mapping which all datasets that belong to a certain class is paired with other datasets from other class for learning as in Figure 1. The number of generated models depending on the number of classes

$$n(n-1)/2 \qquad (1)$$

where n is the number of classes. If the n, is equivalent to 10, so the total of the learned model is 45 according to the mentioned formula. In this method, every class will be paired with other class one-by-one. At the end of the classification (at the testing phase), each classification is given one vote for the winning class. The highest votes will determine which class the test dataset belongs to.

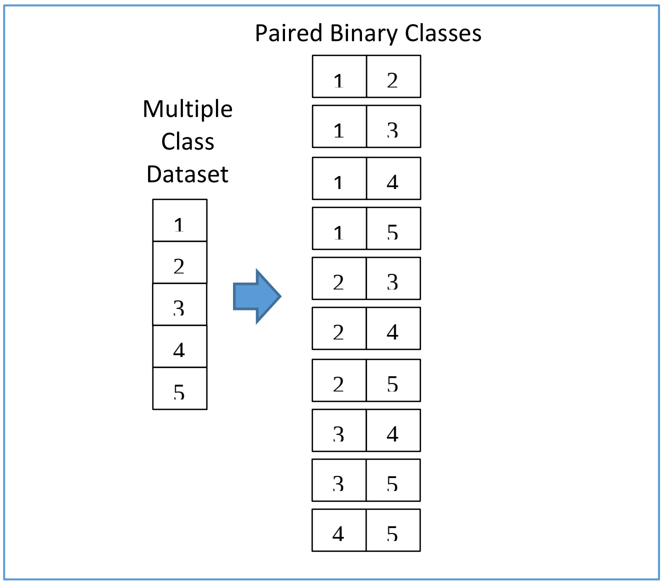

Figure 1: Sample of binarization for OvO when class = 5

C. *One-vs-All (OvA)*

One-vs-all (OvA) is also a paired binary class that involves in the division of n number of classes as in Figure 2. Unlike OvO, OvA produced the same amount of learned models with the number of classes. So, if the number of classes is 10, the number of learned models is also 10.

In this method, every class is paired with the remaining classes. However, this method has the possibility of suffering the imbalance classification if the number of classes is many. This is happens as the number of the training dataset differ tremendously for each learning model [21].

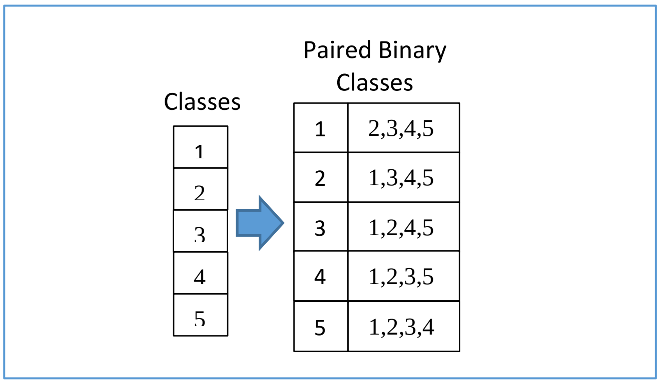

Figure 2: Sample of binarization for OvA when class = 5

D. *Random Correction Code (RCC)*

Random Correction Code is an extension from the original error-correcting output codes (ECOC) [6]. Unlike ECC, RCC is a better choice when the size of the class is big. The extension is able to randomize using pseudorandom number generator of the matrix of the code word at the initial construction [22] which depends on the dataset conditions, seed and width factor as in Figure 3.

At the starting point, the size of the codeword matrix depends on the number of classes and the width factor. The number of class will affect the column size while the width will affect the row size.

The combination of classifiers will be trained depending on the code word so that it adheres to the binary rules. At the testing phase, test dataset will be evaluated for each of the code word classifiers and the one that is the nearest to the test dataset using Hamming distance score will be the winner.

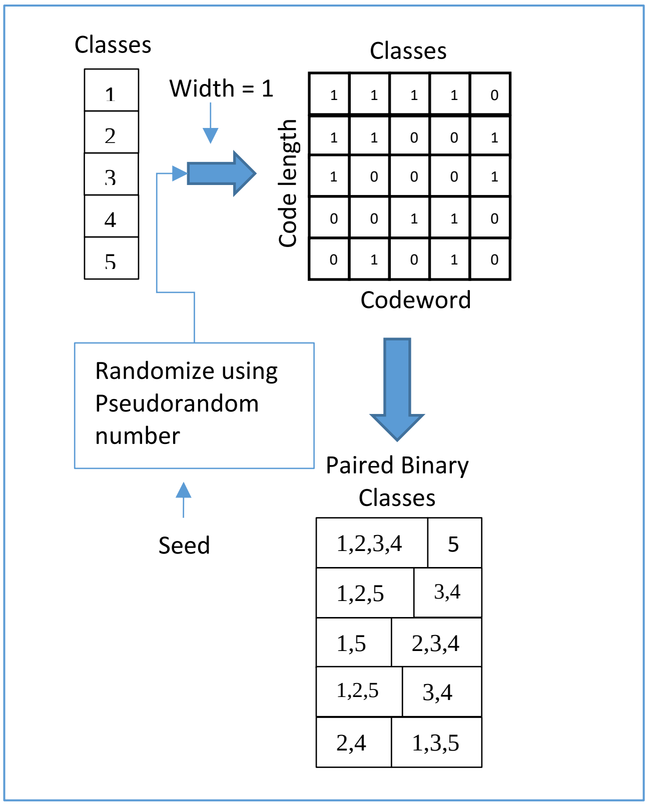

Figure 3: Sample of binarization for RCC when class = 5

IV. Other Based Classifiers

A. *Hoeffding Tree*

Hoeffding tree is a ultra-fast VFDT based decision tree learner which is efficient in mining decision trees from continuously-changing data streams [23]. It works by growing alternative subtree from newer data when the old data become less accurate. When the new data arrives, VFDT will be re-applied in a moving window of the new data.

B. *Random Tree*

Random tree is a tree that takes K randomly chosen attributes at each node [24]. Unlike J48, it has no pruning functionality. Its random engine that based on bagging algorithm depends on the seed in-order to select the attributes. The architecture is a consideration of combining the ideas of single decision tree with Random Forest [25].

C. *Reduced Error Pruning Tree (REPTree)*

REPTree is a fast decision tree learner which builds a decision tree using information gain or variance [24], [25]. It creates multiple trees in different iterations. It is then prunes it using reduce-error pruning with back fitting.

D. *Naïve Bayes*

Naïve Bayes is a simple approach to represent and learn probabilistic knowledge from a given dataset [26]. The term naïve represents its dependency on two assumptions. It assumes that the predictive attributes are conditionally independent give the class.

E. *Stochastic Gradient Descent (SGD)*

Stochastic gradient descent is introduced due to the rapid increase of datasets overtime. It is an advancement from the gradient descent and second order gradient descent which compute the estimation of the gradient on randomly picked sample instead of all [27]. The stochastic process depends on randomly picked samples at each iteration in expecting it behaves like the rest of the dataset despite of the noise. SGD has the capability process the samples on the fly in a deployed system without the needs to remember the visited samples.

F. *Simple Logistic*

Simple logistics or Logistic Model Tree (LMT) is a combination of two classification schemes which are from the linear logistic regression and tree induction [28], [29]. This method builds classification trees with the logistic regression function at the leaves. The fitting process is done by incremental refinement using LogitBoost algorithm at the stagewise level.

G. *Logistic*

Logistic is a function using multinomial logistic regression model with a ridge estimator [30]. It is derived as a restricted maximum likelihood estimator. The used of the ridge estimator is to improve the parameter estimates and to reduce the error made in future prediction.

H. *Fisher's Linear Discriminant Analysis (FLDA)*

In linear discriminant analysis, the idea is to discriminate the classes by finding the best features of the dataset. Its purpose is to find a linear combination of features that characterizes or separates classes in a dataset. It explicitly attempts to model the difference between the data's classes which Principal Component Analysis (PCA) does not. Clearly FLDA deals with independent variables and categorical dependent variable which is the class [31].

V. Experimental Design

In this experiment, data of walking persons are collected while walking on a straight line for a distance approximately 10 meters. Instead of placing the smartphone in the pocket as conducted by [8], [15], [16], the mobile phone is hand held as in the real world situation, it is quite difficult to place the mobile phone in the pocket when the person do no has it [14]. Three types of pose/placement have been captured which are divided into: (1) dataset 1: handheld while the bottom of the smartphone touched the upper abdomen; (2) dataset 2: smartphone is held on palm without touching the body; (3) dataset 3: smartphone is on hand swing.

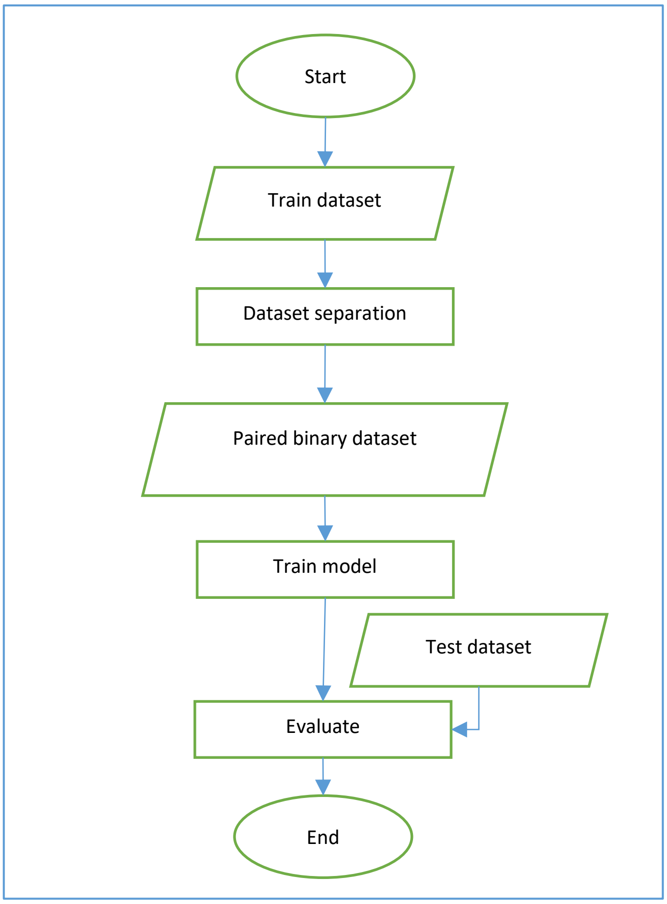

Figure 4: The process flow of the experimentation

At the pre-process stage, linear interpolation and fixed size overlapping sliding window segmentation were employed. For the features extraction, the methods involved are: (1) minimum and maximum value; (2) mean; (3) standard deviation; (4) correlation; (5) root mean square; (6) signal vector magnitude; (7) number of zero crossing of the median and (8) percentile ranks.

For the training phase, the single J48 and multiclass classification mapping method are employed on the training dataset. For the testing dataset, the data remain as original for a real world evaluation. The diagram as in Figure 4. The machine learning classifier used is J48 decision tree which is made as standard among all methods. For the evaluation of the best multiclass classification method, few machine learning algorithm are applied to assess its suitability.

VI. Experimental Result And Discussion

This section presents the experiments involved in this study. A total of 180 samples were used for 3 different positions. For each position, a total of 60 samples were used which are then divided into 30 for training and 30 for testing [14]. The samples for training and testing are came for the same person but the only different is the training set and test set are captured on a different day. Although the total number of people is not many, but it is acceptable just for a simple evaluation [10], [11].

The dataset is undergo for treatment such as preprocessing and features extraction as mentioned in the experimental design. After the features extraction, there are 39 attributes that are used for learning and classification.

Four types of classification mapping were used which are a single classifier, OvA, OvO and RCC. For the RCC, 5 random width factors are implemented. The base classifier is limited to only J48 decision tree.

The experiment is performed on a Windows 10 laptop with Core i7-4710HQ quad core at 2.50 GHz processor with 8 GB of DDR4 RAM.

The evaluation of the multiclass methods are based on the computation time, correct classified instances, incorrectly classified instances, precision, recall, f-measure, ROC area, correct recognition rate, incorrect recognition rate and accuracy of recognition.

Correct classified instances and correct recognition rate are a totally different measurement which correct classified instances consist of evaluation on every single instance from the test set. For the correct recognition rate, the measurement is based on the majority votes derived from the generated confusion matrix. To identify a person, the highest number of instances belong to a particular class is the winner. This method of evaluation also applies to the incorrect classified instances and incorrect recognition rate.

Table 1 shows the number of generated classifiers for each multiclass classifier mapping. OvO contains the most numbers which are 435. This is due to the comparison 1-by-1 for each class.

For the classification evaluation on dataset 1 from Table 2, it can be seen that OvO and RCCs produced the best overall accuracy. For the instances, RCC-5 produced the best score which is 36.7%.

On the dataset 2 from Table 3, the best overall score is OvO followed by RCC-5. For the dataset 3 from Table 4, RCC-5 produced the best accuracy score.

Table 5 depicts the summary or average score for all the multiclass classification methods for all the dataset. It can be seen that the fastest processing speed is the Single J48 due to its simplicity which it is just a single classifier for the training and classification. For the multiclass classification method, OvO is consider as the fastest among all. This is because the dataset are well distributed between the binary classes model. OvA is slower than OvO although the number of classifier is fewer than OvO because OvA methods do suffer high imbalance dataset for each binary models [32]. For the RCC, the higher the factor number, it will getting slower for the training purpose. This is because of the size of the codeword will get larger and the number of classifiers will increase.

For the correct classified instances, RCC-5 produced the best accuracy score if compared to OvO. This is because when a test data is classified, the chances of it fall into a correct class is high unlike OvO. If the RCC factor (width) is higher, the higher chance that the test instance will fall into the correct class. OvA does not perform well because of the severe imbalance class that it suffer.

In the overall accuracy of correct person identification rate, OvO outperforms other methods. The reason of this is OvO performs well in dataset 2 evaluation.

As we observe from the experiment, it can be seen that RCC-5 is consider as the best multiclass classification mapping method as it produce the best overall accuracy score in correct classified instances.

In Table 6, RCC-5 is paired with different types of machine learning classifiers to see its suitability on dataset 1. From the result, it can be seen that J48 and random tree produced the best overall accuracy score. If we look at the correct classified instances, J48 is definitely the clear winner among all.

From the classification evaluation, the multiclass classification mapping accuracy does depend on the dataset. For a better overall learning and classification process, it would be good if the data could be analyzed before deciding any multiclass classification method.

Table 1
Number of classifiers based on the 30 classes (30 people)

Method	**Single J48**	**OvA**	**OvO**	**RCC-1**	**RCC-2**	**RCC-3**	**RCC-4**	**RCC-5**
# of classifiers used	1	30	435	30	60	90	120	150

Table 2
Performance of different classifier mapping method for position 1 (dataset 1 – touch abdomen)

Method	**Single J48**	**OvA**	**OvO**	**RCC-1**	**RCC-2**	**RCC-3**	**RCC-4**	**RCC-5**
Time taken (s)	1.0	8.8	**5.4**	18.5	**36.7**	**55.84**	**74.0**	**92.95**
Correct Classified Instances (%)	19.9	20.2	**30.1**	20.2	**28.4**	**32.5**	**35.2**	**36.7**
Incorrectly classified instances (%)	80.1	79.7	**69.8**	79.8	**71.6**	**67.5**	**64.8**	**63.3**
Precision	0.198	0.216	**0.306**	0.201	**0.282**	**0.318**	**0.346**	**0.357**
Recall	0.199	0.203	**0.302**	0.202	**0.284**	**0.325**	**0.352**	**0.367**
F-measure	0.197	0.205	**0.299**	0.198	**0.277**	**0.317**	**0.345**	**0.357**
ROC area	0.604	0.67	**0.853**	0.726	**0.777**	**0.8**	**0.819**	**0.825**
Correct recognition rate	28	28	**29**	26	**29**	**29**	**29**	**29**
Incorrect recognition rate	2	2	**1**	4	**1**	**1**	**1**	**1**
Accuracy (%)	93.3	93.3	**96.7**	86.7	**96.7**	**96.7**	**96.7**	**96.7**

Table 3
Performance of different classifier mapping method for position 2 (dataset 2 – palm steady)

Method	**Single J48**	**OvA**	**OvO**	**RCC-1**	**RCC-2**	**RCC-3**	**RCC-4**	**RCC-5**
Time taken (s)	3.1	21.7	**13.9**	44.0	86.07	141.67	180.0	224.2
Correct Classified Instances (%)	12.5	11.8	**18.8**	10.9	14.5	16.3	17.2	17.7
Incorrectly classified instances (%)	87.5	88.1	**81.2**	89.1	85.5	83.7	82.8	82.3
Precision	0.125	0.131	**0.187**	0.108	0.139	0.154	0.165	0.173
Recall	0.125	0.118	**0.188**	0.109	0.145	0.163	0.172	0.177
F-measure	0.124	0.111	**0.184**	0.105	0.138	0.154	0.165	0.173
ROC area	0.561	0.605	**0.779**	0.622	0.654	0.685	0.699	0.71
Correct recognition rate	22	13	**26**	14	18	20	22	23
Incorrect recognition rate	8	17	**4**	16	12	10	8	7
Accuracy (%)	73.3	43.3	**86.7**	46.7	60.0	66.7	73.3	76.7

Table 4
Performance of different classifier mapping method for position 3 (dataset 3 – handswing)

Method	Single J48	OvA	OvO	RCC-1	RCC-2	RCC-3	RCC-4	RCC-5
Time taken (s)	0.9	8.2	5.1	18.7	36.1	56.2	**75.1**	**94.1**
Correct Classified Instances (%)	16.2	14.7	21.5	14.4	19.2	21.2	**23**	**25**
Incorrectly classified instances (%)	83.8	85.3	78.5	85.6	80.3	78.8	**77**	**75**
Precision	0.163	0.168	0.215	0.145	0.191	0.212	**0.233**	**0.249**
Recall	0.162	0.147	0.215	0.144	0.192	0.212	**0.23**	**0.25**
F-measure	0.16	0.15	0.211	0.14	0.187	0.206	**0.225**	**0.244**
ROC area	0.58	0.621	0.795	0.672	0.71	0.721	**0.737**	**0.748**
Correct recognition rate	23	22	24	23	23	24	**25**	**25**
Incorrect recognition rate	7	8	6	7	7	6	**5**	**5**
Accuracy (%)	76.7	73.3	80.0	76.7	76.7	80.0	**83.3**	**83.3**

Table 5
Average performance of different classifier mapping method for all positions

Method	Single J48	OvA	OvO	RCC-1	RCC-2	RCC-3	RCC-4	RCC-5
Time taken (s)	**1.67**	12.91	8.13	27.06	52.96	84.57	109.68	137.08
Correct Classified Instances (%)	16.20	15.57	23.47	15.17	20.70	23.33	25.13	**26.47**
Incorrectly classified instances (%)	83.80	84.37	76.50	84.83	79.13	76.67	74.87	**73.53**
Precision	0.16	0.17	0.24	0.15	0.20	0.23	0.25	**0.26**
Recall	0.16	0.16	0.24	0.15	0.21	0.23	0.25	**0.26**
F-measure	0.16	0.16	0.23	0.15	0.20	0.23	0.25	**0.26**
ROC area	0.58	0.63	**0.81**	0.67	0.71	0.74	0.75	0.76
Correct recognition rate	24.33	21.00	**26.33**	21.00	23.33	24.33	25.33	25.67
Incorrect recognition rate	5.67	9.00	**3.67**	9.00	6.67	5.67	4.67	4.33
Accuracy (%)	81.11	70.00	**87.78**	70.00	77.78	81.11	84.44	**85.56**

Table 6
Performance of various classifiers on RCC multiclass classification mapping for position 1

Method	HT	REPTree	Random Tree	J48	NB	FLDA	Logistic	SGD	Simple Logistic
Time taken (s)	4.08	26.67	10.50	**92.95**	4.68	6.61	53.05	64.31	207.90
Correct Classified Instances (%)	15.02	33.40	34.22	**36.7**	11.50	18.23	18.73	14.53	17.46
Incorrectly classified instances (%)	84.98	66.60	65.78	**63.3**	88.50	81.77	81.27	85.47	82.54
Precision	0.157	0.328	0.334	**0.357**	0.123	0.168	0.191	0.173	0.173
Recall	0.150	0.334	0.342	**0.367**	0.115	0.182	0.187	0.145	0.175
F-measure	0.130	0.329	0.333	**0.357**	0.083	0.159	0.164	0.118	0.146
ROC area	0.715	0.837	0.811	**0.825**	0.714	0.772	0.778	0.760	0.772
Correct recognition rate	16	28	29	**29**	6	21	22	19	19

Incorrect recognition rate	14	2	1	**1**	24	9	8	11	11
Accuracy (%)	53.33	93.33	96.67	**96.67**	20.00	70.00	73.33	63.33	63.33

VII. CONCLUSION

In this study, we presented a study on three types of multiclass classification mapping method for classifying dataset that contains many classes on a handheld based smartphone gait identification. The methods are OvA, OvO and RCC with 5 types of random width factor. As a benchmark, a single classifier J48 is also used to classify the gait signal. From the result, it is proven partially (OvA is considered bad if the number of classes is high) that using multiclass classification mapping does increase the accuracy rate in the overall classification hence can be the stepping stone for a bigger scope of classes in any field or situation. The accuracy of the RCC could be further improved when the width factor is increased but the time for learning phase will be increased and computational cost would be higher. Besides that, due to the high requirements of the computer resources, major machine learning classifiers such as MLP and SVM could not be implemented.

REFERENCES

[1] M. W. Whittle, *Gait analysis: an introduction*. Butterworth-Heinemann, 2014.

[2] J. Favre, R. Aissaoui, B. M. Jolles, J. A. de Guise, and K. Aminian, “Functional calibration procedure for 3D knee joint angle description using inertial sensors,” *J. Biomech.*, vol. 42, no. 14, pp. 2330–2335, 2009.

[3] T. Liu, Y. Inoue, and K. Shibata, “Development of a wearable sensor system for quantitative gait analysis,” *Measurement*, vol. 42, no. 7, pp. 978–988, 2009.

[4] J. Klucken, J. Barth, P. Kugler, J. Schlachetzki, T. Henze, F. Marxreiter, Z. Kohl, R. Steidl, J. Hornegger, B. Eskofier, and others, “Unbiased and mobile gait analysis detects motor impairment in Parkinson’s disease,” *PLoS One*, vol. 8, no. 2, p. e56956, 2013.

[5] A. Fernández, M. J. Del Jesus, and F. Herrera, “Multi-class imbalanced data-sets with linguistic fuzzy rule based classification systems based on pairwise learning,” in *Computational Intelligence for Knowledge-Based Systems Design*, Springer, 2010, pp. 89–98.

[6] T. G. Dietterich and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes,” *J. Artif. Intell. Res.*, vol. 2, pp. 263–286, 1995.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[8] M. Derawi and P. Bours, “Gait and activity recognition using commercial phones,” *Comput. Secur.*, vol. 39, no. 2, pp. 137–144, 2013.

[9] C. Nickel and C. Busch, “Classifying accelerometer data via hidden Markov models to authenticate people by the way they walk,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 28, no. 10, pp. 29–35, 2013.

[10] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, “User Verification Leveraging Gait Recognition for Smartphone Enabled Mobile Healthcare Systems,” *IEEE Trans. Mob. Comput.*, vol. 14, no. 9, pp. 1961–1974, 2015.

[11] T. Hoang, T. Nguyen, C. Luong, S. Do, and D. Choi, “Adaptive cross-device gait recognition using a mobile accelerometer,” *J. Inf. Process. Syst.*, vol. 9, no. 2, pp. 333–348, 2013.

[12] X. Dong, L. Qian, Y. Guan, L. Huang, Q. Yu, and J. Yang, “A multiclass classification method based on deep learning for named entity recognition in electronic medical records,” in *Scientific Data Summit (NYSDS), 2016 New York*, 2016, pp. 1–10.

[13] M. N. I. Qureshi, B. Min, H. J. Jo, and B. Lee, “Multiclass classification for the differential diagnosis on the ADHD subtypes using recursive feature elimination and hierarchical extreme learning machine: structural MRI study,” *PLoS One*, vol. 11, no. 8, p. e0160697, 2016.

[14] A. R. Abdul Raziff, M. N. Sulaiman, N. Mustapha, and T. Perumal, “Smote and OVO Multiclass Method for Multiple Handheld Placement Gait Identification on Smartphone’s Accelerometer,” *J. Eng. Appl. Sci.*, vol. 12, no. 2, pp. 374–382, 2017.

[15] B. Sun, Y. Wang, and J. Banda, “Gait Characteristic Analysis and Identification Based on theiPhone’s Accelerometer and Gyrometer,” *Sensors*, vol. 14, no. 9, pp. 17037–17054, 2014.

[16] T. Hoang, D. Choi, and T. Nguyen, “Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment scheme,” *Int. J. Inf. Secur.*, vol. 14, no. 6, pp. 549–560, 2015.

[17] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, “The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication,” *Pattern Recognit.*, vol. 47, no. 1, pp. 228–237, 2014.

[18] J. Frank, S. Mannor, J. Pineau, and D. Precup, “Time Series Analysis Using Geometric Template Matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 740–754, 2013.

[19] S. L. Salzberg, “C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993,” *Mach. Learn.*, vol. 16, no. 3, pp. 235–240, 1994.

[20] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[21] J. A. Gualtieri and R. F. Cromp, “Support vector machines for hyperspectral remote sensing classification,” in *The 27th AIPR Workshop: Advances in Computer-Assisted Recognition*, 1999, pp. 221–232.

[22] G. James and T. Hastie, “The error coding method and PICTs,” *J. Comput. Graph. Stat.*, vol. 7, no. 3, pp. 377–387, 1998.

[23] G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 97–106.

[24] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[25] S. Kalmegh, “Analysis of WEKA data mining algorithm REPTree, Simple CART and RandomTree for classification of Indian news,” *Int. J. Innov. Sci. Eng. Technol.*, vol. 2, no. 2, pp. 438–446, 2015.

[26] G. H. John and P. Langley, “Estimating continuous distributions in Bayesian classifiers,” in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, 1995, pp. 338–345.

[27] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT’2010*, Springer, 2010, pp. 177–186.

[28] M. Sumner, E. Frank, and M. Hall, “Speeding up logistic model tree induction,” in *European Conference on Principles of Data Mining and Knowledge Discovery*, 2005, pp. 675–683.

[29] N. Landwehr, M. Hall, and E. Frank, “Logistic model trees,” *Mach. Learn.*, vol. 59, no. 1–2, pp. 161–205, 2005.

[30] S. Le Cessie and J. C. Van Houwelingen, “Ridge estimators in logistic regression,” *Appl. Stat.*, pp. 191–201, 1992.

[31] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, “Fisher discriminant analysis with kernels,” in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, 1999, pp. 41–48.

[32] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, “Label ranking by learning pairwise preferences,” *Artif. Intell.*, vol. 172, no. 16, pp. 1897–1916, 2008.