IOP Conf. Series: Materials Science and Engineering **318** (2018) 012046 doi:10.1088/1757-899X/318/1/012046

An Improvement To The k-Nearest Neighbor Classifier For ECG Database

Haryati Jaafar[1,*], Nur Hidayah Ramli[1], Aimi Salihah Abdul Nasir[1]

[1]Faculty of Engineering Technology, Universiti Malaysia Perlis, UniCITI Alam Campus, Sg. Chuchuh, 02100 Padang Besar, Perlis, Malaysia

[*]haryati@unimap.edu.my

Abstract. The k nearest neighbor (kNN) is a non-parametric classifier and has been widely used for pattern classification. However, in practice, the performance of kNN often tends to fail due to the lack of information on how the samples are distributed among them. Moreover, kNN is no longer optimal when the training samples are limited. Another problem observed in kNN is regarding the weighting issues in assigning the class label before classification. Thus, to solve these limitations, a new classifier called Mahalanobis fuzzy k-nearest centroid neighbor (MFkNCN) is proposed in this study. Here, a Mahalanobis distance is applied to avoid the imbalance of samples distribition. Then, a surrounding rule is employed to obtain the nearest centroid neighbor based on the distributions of training samples and its distance to the query point. Consequently, the fuzzy membership function is employed to assign the query point to the class label which is frequently represented by the nearest centroid neighbor Experimental studies from electrocardiogram (ECG) signal is applied in this study. The classification performances are evaluated in two experimental steps i.e. different values of k and different sizes of feature dimensions. Subsequently, a comparative study of kNN, kNCN, FkNN and MFkCNN classifier is conducted to evaluate the performances of the proposed classifier. The results show that the performance of MFkNCN consistently exceeds the kNN, kNCN and FkNN with the best classification rates of 96.5%.

1. Introduction

Classification processes have been extensively studied in the machine learning literature, such as face classification, speech classification, handwriting character classification and others. These processes are divided into unsupervised and supervised classification [1, 2]. In unsupervised classification, a clustering technique is applied to find the similarity of samples before the samples are classified into a small number of homogenous or same groups. For a supervised classification, the classification is tackled through two classifiers, namely parametric and nonparametric classifiers. The parametric classifier is based on the assumption of given samples that are normally distributed. Some parametric classifiers include the support vector machine (SVM), naïve Bayes and decision tree. In contrast, the non-parametric classifiers do not make any assumptions on the samples' distribution. These classifiers include the k-nearest neighbor (kNN), multilayer perceptron (MLP), artificial neural network (ANN) and Euclidean distance (ED) [3].

This research focuses on the kNN which is under the supervised and non-parametric category. This classifier was introduced by Cover and Hart in 1976 [4]. Consequently, it is regularly used in practice since this classifier is simple to implement and quite straightforward in the classification process.

According to the kNN rule, the classification is based on a majority voting process. In this process, a testing sample or query point is assigned to the class label which has been represented by the majority label of its k-nearest neighbors. The k-nearest neighbors are determined from the minimum distance between the query point and training samples. Nevertheless, the drawback of kNN is that the distributions of samples are not taken into account in this classifier. There are different types of samples distribution either around or aside from the query point. Hence, the nearest neighbors may not have the symmetry around the query point if the samples are not homogenous or of the same class in its neighborhood [5]. Another major problem is that the kNN rule is no longer optimal if the number of training samples is limited. This is because the limited number of training samples can affect the region of neighbourhood to be smaller. Thus, the kNN rule will select the nearest training samples within the region. However, the nearest neighbour classes in the region may be unreliable which misclassify the query point [6].

In order to overcome the problems occurring in kNN, studies on the concept of neighborhood are proposed. According to Chaudhuri [5], the concept of neighborhood should be such that (a) the neighbors are as near to the query point as possible and (b) the neighbors sit as symmetrically around the query point as possible. To tackle the problem, Chaudhuri [5] proposes a simple and intuitively appealing definition of neighborhood called the nearest centroid neighborhood (NCN) and nearest median neighborhood (NMN). In the NCN concept, the idea of the neighbourhood is based on two criteria. Firstly, by the distance criterion where the k neighbours of a query point must be as near as possible. Secondly, by the symmetry criterion, their centroid must also be as close to the query point as possible. For the second approach which is the NMN, the median point is used instead of centroid to define the neighbors. Following the earlier work, Sanchez et al., [3] introduce a surrounding neighborhood (SN) to employ the distances and distributions of training samples before the classification process.

On the other hand, an extension of the kNN using the fuzzy rule has been proposed by Keller et al., [7]. In the fuzzy rule-based system, the fuzzy kNN (FkNN) assigns a fuzzy membership for the query point rather than assigning the query point to the class label. The fuzzy membership is determined based on the class weighted. The class weighted is inversely proportional to the distance between training samples and the query point [13]. The highest fuzzy membership value is considered as a winner for that query point.

Motivated by the capability of the NCN and FkNN to handle the kNN problems, an extensive improvement classifier of kNN called a Mahalanobis fuzzy k-nearest centroid neighbor (MFkNCN) classifier is proposed in this paper. This classifier is divided into two parts. In the first part, an adaptive technique by using Fuzzy Inference System (FIS) is introduced to reduce the size of the training set aiming for both reducing complexity and avoiding the outliers when using the entire training set. A Mahalanobis distance is employed in these algorithms.

Subsequently, in second part, a surrounding-fuzzy based rule is employed. This rule is adapted from the SN rule and fuzzy rule applied in the NCN and FkNN, respectively. The surrounding rule is used to ensure that the training samples are distributed sufficiently in the region of neighbourhood with the nearest neighbours located around the query point. Consequently, the fuzzy-based rule is employed to solve the ambiguity of the weighting distance between the query point and its nearest neighbors.

In order to determine the effectiveness of propsed classifier, experimental studies from an ECG signal database is employed in this study. This database is obtained freely from public heart sound database, assembled for an international competition, the PhysioNet/Computing in Cardiology (CinC) Challenge 2016. The archive comprises nine different heart sound databases sourced from multiple research groups around the world.

This paper is outlined as follows. The proposed classifier, MMFkNCN and the rationale to use FIS and Mahalanobis distance measure is presented in Section 2. The surrounding-fuzzy based rule is explained in Section 3 and Section 4 discusses the experimental results and discussion. Finally, conclusions are summarized in Section 5.

2. Part 1: Fuzzy Inference System

The goal in this section is to investigate an iterative way to reduce the storage requirement and select the best candidate from the training samples without degrading performance of the MMFkNCN. They are achieved by employing the outlier detection based on fuzzy-rule. Here, two algorithms which are the Weighted Similarity Function (WSF) and triangle inequality are considered. A weighted similarity function is used to calculate the distance (or similarity) of a query point to the training samples while triangle inequality is implemented to search for the threshold for the outlier detection. The training samples that fell within a threshold are defined as the candidate samples. The unselected training sample is called as the outlier where it is defined as a noisy sample that is excluded during the second part operation.

2.1. Mahalanobis distance

In most of the nearest neighbor classifier, the Euclidean distance is conventionally used to measure the distance between the query point and training samples. It is appropriate for all numeric training samples with similar length. However, the class distribution of the data in the training samples is imbalanced when one of the classes is more under-represented than the other classes [3]. Hence, a severe bias can occur when using the Euclidean distance as the training samples are assumed as balanced distribution in most cases.

Therefore, the Mahalanobis distance is introduced in this study instead of measuring the training samples using the Euclidean distance. This distance is a well-known criterion which depends on estimated parameters of the multivariate distribution. Given a query point *y*, a set of training samples $T = \{x_j\}_{j=1}^{N}$ with the label patterns of $\{y_i \mid i = 1, 2, ..., n\}$ where *N* is the number of training samples, x_j is the training sample and *n* is the number of class. Then, the sample mean vector by $\overline{x}_N$ and the sample covariance matrix by C_N ,

$$C_N = \frac{1}{N-1}\sum_{j=1}^{n}(y - \overline{x}_N)(y - \overline{x}_N)^T \tag{1}$$

The Mahalanobis distance for each multivariate data point $\{y_i \mid i = 1, 2, ..., n\}$ is given as :

$$M(y, x_j) = \left(\sum_{j=1}^{n}(y - \overline{x}_N)^T C_N^{-1}(y - \overline{x}_N)\right)^{1/2} \tag{2}$$

2.2. Weighted similarity function

The WSF is applied to efficiently find the similarity of the query point and the training samples. This is done by assigning the weight to each query point in the training set. The similarity is important to estimate and check the similarity and dissimilarity values between them. Technically, the similarity values would increase and dissimilarity values would decrease when both the query point and training sample became more similar. In the WSF, the computation of similarity depended on weight coefficients $w(y,x_j)$. As the training sample has larger $w(y,x_j)$, it became more similar and closer to the query point [8].

In order to apply the WSF, the information of a query point *y* and a set of training sets $T = \{x_j\}_{j=1}^{N}$ are firstly determined. Then the Mahalanobis distance, $M(y,x_j)$ between the query point and the training samples in Equation (2) is computed. Subsequently, the WSF based on weight coefficients, $w(y,x_j)$ is determined by modifying the distance function in Equation (2) to form the following equation:

$$w(y, x_j) = \frac{1 - M(y, x_j)}{M_{\max}} \tag{3}$$

where $w(y,x_j)$ is in the range [0,1], M_{max} is the maximum possible distance between two training samples in the feature space.

2.3. Triangle inequality

To define whether the sample is either a candidate sample or outlier, the triangle inequality is employed in this study. Considering x represented the training sample, y as a query point and z is the reference point, the reference point is the furthest training sample from the query point that is obtained from Equation (2). The distances are sorted in ascending order before determining the reference point. The condition for applying the triangle inequality technique is that the distance between two objects should not be less than the difference in the distances to any objects [9]. Specifically, if the distance between the query point and the training samples satisfies Equation (3), the following inequality would always true.

$$M(y,x_j) \le M(y,z) + M(x_j,z) \tag{4}$$

By comparing the training sample and query point to a reference point, an upper bound is obtained as:

$$\left| M(y,z) - M(y,x_j) \right| \le M(x_j,z) \le M(y,z) + M(y,x_j) \tag{5}$$

According to Equations (4) and (5), the triangle inequality and upper bound depended on the distance between the training sample to the reference point, $d(x_j,z)$. In order to avoid calculation of $d(x_j,z)$, a lower bound is needed such that $M(x_j,z) \ge M(y,x_j)$.

Let:

$$M(y,x_j) \le \frac{1}{2} M(y,z) \tag{6}$$

Hence, Equation (4) turned to be:

$$2M(y,x_j) \le M(y,z) + M(x_j,z) \tag{7}$$

Thus, Equation (7) proved that:

$$M(y,x_j) \le M(x_j,z) \qquad .(8)$$

Equation (6) is rearranged and as a result:

$$M(x_j,z) \ge M(y,x_j) - M(y,z) \tag{9}$$

Assuming $M(x_j,z) \ge 0$, the lower bound is obtained as:

$$M(x_j,z) \ge \max\left\{0, M(y,z) - M(y,x_j)\right\} \tag{10}$$

and the distance $M(x_j,z)$ can be avoided. Since $M(y,x_j) \le M(x_j,z)$, the training sample is selected as the candidate sample after the Equation (6) is rewritten as:

$$\frac{M(y,x_j)}{M(y,z)} \le \frac{1}{2} \tag{11}$$

The process of traversing a list of the candidate samples continued until it met the lower and upper bound criterion.

2.4. Design on FIS

There are three main stages to compute the output of the FIS i.e. fuzzification process, rule extraction and defuzzification process. The Mamdani model was also adopted in this study since it effectively modelled the common-sense rules on fuzzy variables.

The fuzzification is the process of finding the degree of membership of a value in a fuzzy set. In this process, the input variables are converted from crisp values to fuzzy values by using a set of input Membership Functions (MF) In this study, the Gaussian curve is chosen to determine the membership values. In this process, the weight coefficients, $w(y,x_j)$ from Equation (3) and the ratio of triangle inequality from Equation (11) are employed as the input variables and the MFs and the outlierness is assigned as an output variable. The quantization level [10] for weight coefficients, $w(y,x_j)$, $\mu_A(x)$, triangle inequality, $\mu_B(x)$ and outlierness, $\mu_C(x)$ are summarized in Table 1.

Table 1. Quantization level of A and B fuzzy sets

Quantization level	A (weight coefficients)	B (ratio of triangle inequality)	C (outlierness)
V1=0	Small	Narrow	High
V2 = 0.5	Medium	Moderate	Intermediate
V3 = 1	Large	Wide	Low

As the MFs of input and output variables are defined in the FIS, the rule base is then evaluated. The input membership values such as the weighting factors are used by the rules to determine their influence on the set of fuzzy outputs in the final output conclusion. In this study, nine rules are used to characterize fuzzy rules and they are shown in Table 2.

Table 2. Rule extracted from A and B for outlier detection

Rules	Inputs		Output
	A (weight coefficients)	*B* (ratio of triangle inequality)	*C* (outlierness)
Rule 1	Small	Narrow	Intermediate
Rule 2	Small	Moderate	Not high
Rule 3	Small	Wide	High
Rule 4	Medium	Narrow	Not low
Rule 5	Medium	Moderate	Intermediate
Rule 6	Medium	Wide	Not high
Rule 7	Large	Narrow	Low
Rule 8	Large	Moderate	Not low
Rule 9	Large	Wide	Intermediate

In the defuzzification process, the fuzzy output values are converted into a single crisp value or final decision. The COG was applied and the crisp value obtained is:

$$y_i = \frac{\int_y y.\mu_C(y_i)dy}{\int_y \mu_C(y_i)dy} \tag{12}$$

where *Yi* is a centroid of fuzzy set *C* or outlierness.

The defuzzified output of the fuzzy procedure is influenced by the value of weight coefficient and the ratio of the triangle inequality. For example, if the value of weight coefficient is large and ratio of the triangle inequality is narrow, the outlierness was low hence the training sample is accepted as a candidate training sample.

3. Part 2: Surrounding-fuzzy Based Rule

In the second part, the surrounding-fuzzy based rule is proposed where this rule is modified from the surrounding rule and fuzzy rule that is applied in the kNCN and FkNN, respectively. The main objective of this classifier is to optimize the performance results while considering the surrounding-fuzzy based rule which are

i) the k centroid nearest neighbors should be close to the query point as possible and located symmetrically around the query point.

ii) the query point is classified by taking the fuzzy membership values into account.

In this rule, the candidate samples that obtained from Part 1 is used to define the k neighbors. A set of candidate samples $T_s = \{x_s\}_{s=1}^{Q}$ with the label pattern of $c_1, c_2, \ldots c_n$, is given where Q is the number of training samples, x_s is the candidate sample and n is the number of class. The distance between the query point and the candidate samples is found by using Equation (13).

$$d(y,x_s) = \|y - x_s\|_{L2} \tag{13}$$

A set of k nearest centroid neighbor $X^{NCN} = \{x_r^{NCN}\}_{r=1}^{k}$ is considered where k is the number of nearest centroid neighbors. By applying Equation (13), the first centroid neighbor is determined by selecting the training sample that is located the nearest to the query point. The first nearest centroid neighbor is set as x_1^{NCN}. The subsequent nearest centroid neighbors for r=2,…,k are determined by calculating the centroid position of all the training samples except the training sample that has been selected as the nearest centroid neighbor. The centroid of a set of training samples, x_{rj}^{c} can be defined as:

$$x_{rj}^{c} = \frac{\left(\sum_{r=1}^{r-1} x_r^{NCN}\right) + x_j}{r} \tag{14}$$

The k nearest centroid neighbor is determined by computing the shortest distance between the centroid of a set of a training samples and the query point. It is given as:

$$x_r^{NCN} = \min \|(y - x_{rj}^{c})\|_{L2} \tag{15}$$

The process is repeated by determining the centroid between the training samples and previous nearest centroid neighbors for the value of r=3.

After the k nearest centroid neighbors has been determined, the next step is to assign a fuzzy membership for the k nearest centroid neighbors to class labelled as y rather than voting the k nearest centroid neighbors as a majority. By assigning the fuzzy membership, the MFkNCN classifier become more advantageous as it is able to provide the next potential secondary structural class if the first one is wrongly predicted. This process is done by determining the weight that is inversely proportional to the distance between x_r^{NCN} and y. The fuzzy membership found is:

$$u_i^{NCN}(y) = \frac{\sum_{r=1}^{k} u_{ir}\left(1/\left\| y - x_r^{NCN} \right\|^{2/(m-1)}\right)}{\sum_{r=1}^{k} 1/\left\| y - x_r^{NCN} \right\|^{2/(m-1)}} \tag{16}$$

where i = 1,2,...,C, with C as the number of classes and u_{ir} as the membership degree of k nearest centroid neighbors.

The value of the fuzzy strength parameter is set at 2. This was because the fuzzy membership values are proportional to the inverse square of the distance. When it is set at 2, it leads to an optimal result in the classification process. The constraint fuzzy membership is given as [7]:

$$u_{ir}(x_k) = \begin{cases} 0.51 + 0.49(n_r / k) & r = i \\ 0.49(n_r / k) & r \neq i \end{cases} \tag{17}$$

where n_r denoted the number of rth nearest centroid neighbors.

After calculating all memberships for a query point, it is assigned to the class of the highest membership value as shown in Equation (17).

$$C(y) = \arg\max(u_i^{NCN}(y)) \tag{18}$$

The process of the MFkNCN classifier during the searching stage is then repeated for a new query point until all query points are successfully classified [11].

4. Experimental results

4.1. Data acquisition

In this study, the Shiraz University adult heart sounds database (SUAHSDB) is employed. This database was constructed using recordings made from 79 healthy subjects and 33 patients (total 69 female and 43 male, aged from 16 to 88 years). During the recording, the subjects were asked to relax and breathe normally during the recording session. The database consists of 114 recordings (81 normal recordings and 33 pathological recordings). The recording length varied from approximately 30s–60s. The sampling rate was 8000 Hz with 16 bit quantization except for three recordings at 44 100 Hz and one at 384000 Hz. The data were recorded in wideband mode of the digital stethoscope, with a frequency response of 20 Hz–1 kHz [12].

4.2. Experimental setup

A selected signal from 79 healthy subjects with 79 recordings is applied for experiment. The principle component analysis with autocorrelation method is selected as a feature extraction due to the fact that the feature is more robust to noise compared to other feature extractions [12]. Subsequently, a comparative study of kNN, kNCN, FkNN and MFkNCN classifiers is conducted to evaluate the performances of the classifiers. Two experiments were conducted in this study, where in the first experiment, performance evaluations with different values of k were observed. The values of k ranging was set from 1 to 10 while the size of feature dimension was fixed to 64. In this experiment, the training samples were set to 30 and the testing data set to 49. In the second experiment which is performance evaluation based on different sizes of feature dimensions, four sizes of feature dimensions i.e. 10, 16, 32 and 64 are computed.

The proposed methods have been implemented in Matlab R2010 (b) and have been tested in Intel Core i5, 2.1GHz CPU, 6G RAM and Windows 7 operating system. In all experiments, the performance was evaluated based on the classification accuracy (CA) which is calculated as;

$$CA = \frac{N_C}{N_T} \times 100\% \tag{19}$$

where Nc is the number of syllables which is recognized correctly and N_T is the total number of test syllables.

4.3. Performances evaluation

Figure 1 illustrates the performance of four classifiers in term of different values of k. It shows that the kNN gives the poorest classification performance as compares to kNCN, FkNN and MFkNCN. In addition, the classification gradually decreases as the values of k increases. Although the highest classification accuracy of kNN is 94.59% at k=1 and 2, the performance of the kNN is still poor compared to the kNCN, FkNN and MFkNCN. For the kNCN, it is observed that the performance of the kNCN increases as the number values of k varies from 1 to 7 with the highest classification accuracy 96.49% at k=7. Moreover, this classifier is able to perform better than the kNN and FkNN in most of the values of k. On the other hand, the highest accuracy for the FkNN is 95.88% at k=8. The results show that the FkNN outperforms the kNN at every values of k. Nevertheless, this classifier performs poorly than the kNCN and MFkNCN in most of the values of k. For the MFkNCN, the results show that the performance of the MFkNCN consistently exceeds that of the kNN and FkNN. Although the performance of MFkNCN is lower than kNCN at k=7 and 10, the MFkNCN is able to achieve the better performance than kNCN almost every values of k with the best classification performance is obtained at k=9 with 96.55%. However, all of the classifiers show an instable trend when the number of k is increased. This is because some of the training samples from different classes have a very similar characters and these training samples are called overlapping samples. Thus, the misclassification often occurs near class boundaries where overlapping usually occurs as well [10].

For the second experiment, the value of k was fixed to 9 based on the highest accuracy that achieved in MFkNCN from the previous experiment. Figure 2 shows the classification performance at different sizes of feature dimensions for each classifier i.e. kNN, kNCN, FkNN and MFkNCN. As it can see in Figure 2, the performance of kNN provides worst performance, yielding 90.8% and 92.09% classification accuracy for the highest and lowest accuracy at 10 and 64 feature dimensions, respectively. The results show the kNCN outperforms the kNN and FkNN where the highest average accuracy achieved is 96.19%. For the FkNN, the performance of this classifier is better than kNN with the highest classification accuracy is 95.29%. It can be observed that the performance of the MFkNCN consistently exceeds of the kNN and FkNN at all sizes of feature dimensions. In 16 feature dimension, the kNCN slightly outperform the MFkNCN on this dataset. However, the performance of MFkNCN varies more in other feature dimensions. The best classification performance is achieved at 64 size of feature dimension with 96.55%.

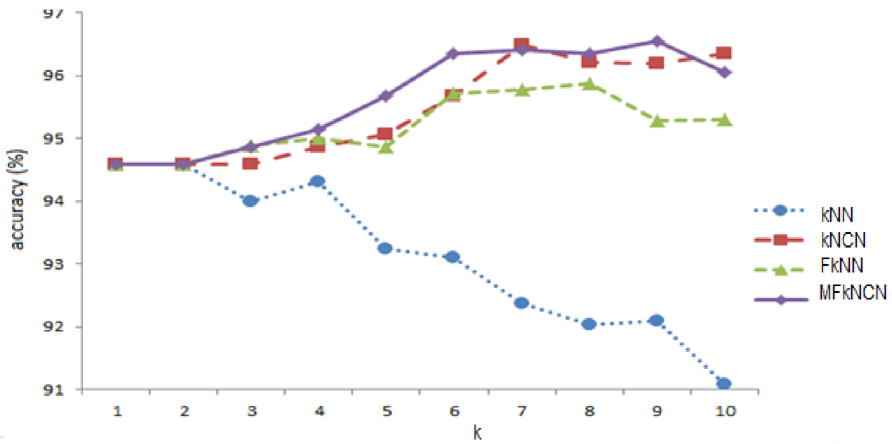

Figure 1. Performances of kNN, kNCN, FkNN and MFkNCN based on different values of k

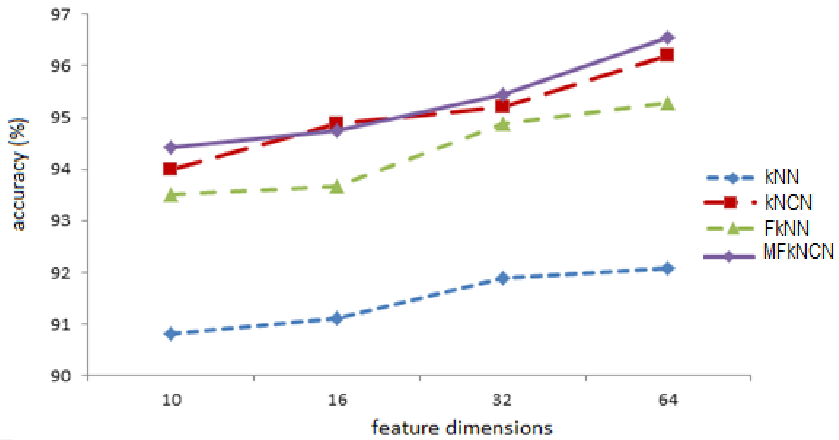

Figure2. Performances of kNN, kNCN, FkNN and MFkNCN based on different sizes of feature dimensions

5. Conclusion

In this paper, an empirical work of the centroid neighborhood and fuzzy-rule based system with Mahalanobis distance called a a Mahalanobis fuzzy k-nearest centroid neighbor (MFkNCN) has been proposed and successfully implemented. The proposed classifier aims at exploiting the strength of the centroid neighbourhood while solving the ambiguity of the weighting distance between the query point and its nearest neighbours by using Mahalanobis distance. The proposed classifier firstly computed the k nearest centroid neighbourhood for each class separately. Then, the fuzzy membership is constructed to assign the query point to the right class label. In this paper, the proposed classifier has been compared with the kNN, kNCN and FkNN to validate the effectiveness of the MFkNCN. A series of experiments, based on different values of k and different sizes of feature dimensions have been performed to determine the competence of the proposed classifier. It was found that the classification accuracies were up to 92.09%, 96.19%, 95.29% and 96.55% for kNN, kNCN, FkNN and MFkNCN, respectively. Meanwhile, for the palm prints dataset, the kNN, kNCN, Results indicate that the MFkNCN provides the best classification among the classifiers.

Acknowledgement
The authors would like to express their gratitude for the financial support provided by Universiti Malaysia Perlis Short Term Grant (9001-00500).

References

[1] Sheskin D J 2011 *Handbook of Parametric and Nonparametric Statistical Procedures,* (Chapman & Hall)
[2] Kumar Y and Sahoo G 2012 Analysis of Parametric & Non Parametric Classifiers for Classification Technique using WEKA *International Journal Information Technology and Computer Science* Volume **7** pp 43-49
[3] Sanchez J S, Pla F and Ferri F J 1997 On the use of neighbourhood-based non- parametric classifiers *Pattern Recognition Letters* Volume 18 pp 1179–1186
[4] Cover T and Hart P 1967 Nearest neighbour pattern classification *IEEE Trans on Information Theory* Volume 13 Number 1pp 21-27
[5] Chaudhuri B B 1996 A new definition of neighbourhood of a point in multi-dimensional space *Pattern Recognition Letters* Volume 17 Number 1 pp 11-17
[6] Imandoust S B and Bolandraftar M 2013 Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background. *Int. Journal of Engineering Research and Application* Volume 3 Number 5 pp 605-610
[7] Keller J M Gray, M R and Givens, J A 1985 A fuzzy k-nearest neighbor *algorithm IEEE Trans.Syst., Man, Cybern. SMC* Volume 15 Number 4 pp 580-585
[8] Triguero I, Derrac J, Garcia S and Herrera F 2012 A taxonomy and experimental study on prototype generation for nearest neighbor classification *IEEE Trans. on Sys, Man, and Cybernetics, Part C: Apps & Reviews* Volume **42** Number 1 pp 86-100
[9] Berman A and Shapiro L G 1998 Selecting good keys for triangle-inequality-based pruning algorithms *IEEE International Workshop on Content-Based Access of Image and Video Database* pp 12-19
[10] Jaafar H , Ibrahim S and Ramli D A 2015 Robust and fast computation touch-less palm print recognition system using LHEAT and IFkNCN classifier Computational *Intelligence and Neuroscience* pp 1-17
[11] Jaafar H, Mukahar N and Ramli D A 2016 A methodology of nearest neighbor: Design and comparison of biometric image database *IEEE Student Conference on Research and Development (SCOReD)* pp 1-6
[12] Liu C, Springer D, Li Q, Moody B, Juan R A, Chorro F J, Castells F, Roig J M, Silva I, Johnson A E and Syed Z 2016 An open access database for the evaluation of heart sound algorithms *Physiological Measurement* Volume 37 Number 12 pp 2181