

# A Comparative Study of Various Classifiers for Character Recognition on Multi-script Databases

Jamimamul Bakas  
Dept. of Computer Sc. & Engg.  
Aliah University  
Kolkata, India

Mahabub H. Mahalat  
Dept. of Computer Sc. & Engg.  
Aliah University  
Kolkata, India

Ayatullah F. Mollah  
Dept. of Computer Sc. & Engg.  
Aliah University  
Kolkata, India

## ABSTRACT

Classification through learning from examples is extensively applied to character recognition from last three decades. Considerable improvements in terms of classification accuracies have been made using various classification methods. But, comparison of various classifiers for the same character dataset research is not exhaustive. This paper investigates the recognition performance of support vector machine (SVM) with various kernels, multi-layer perceptron (MLP), k-nearest neighbors (kNN), naive Bayes and minimum distance classifiers for character recognition on multi-script databases viz. Arabic, Oriya and Bengali. It is found that MLP performed the best for Oriya (95.20%) and Bengali (95.10%) datasets, and SVM with radial basis function (RBF) kernel performs the best for Arabic (96.70%) dataset. Among other classifiers, kNN is giving relatively better results. In all cases, minimum distance classifier gives the worst performance. In total, MLP followed by SVM RBF kernel is found to be the most efficient among all classifiers included in this study.

## Keywords

Bayes Classifier, Minimum Distance, K-Nearest Neighbors, Multi Layer Perceptron, Support Vector Machine, Linear Kernel, Quadratic Kernel, Polynomial Kernel, Radial Basis Function Kernel, Comparative Study

## 1. INTRODUCTION

Development of artificial cognition power for machines is still an active research problem. Pattern recognition is an important sub-field of artificial cognition. In this domain, features are extracted from objects to classify them into some predefined classes using techniques called as classifiers. Hence, the performance of a classifier plays crucial role in pattern classification. Plenty of classifiers have been developed for this purpose. So, a comparative analysis is congenial to choose the appropriate classifier for a certain classification problem. Moreover, a classifier may not perform equally well for all types of objects. So, domain specific comparison of classifiers is required.

Handwritten optical character recognition is an important problem in this domain. Unlike printed characters, handwritten characters extensively differ in size, shape and style from person to person and even from time to time for the same person. A character recognition engine must recognize most of the prevalent handwriting styles.

Like any image, visual characters are subject to spoilage due to noise. Noise consists of random changes to a pattern, particularly near the edges. A character with much noise may be interpreted as a completely different character by a computer program. There are no hard-and-fast rules that define the appearance of a visual character. Hence, rules need to be heuristically deduced from the samples.

Many applications such as license plate recognition, smart card processing system, automatic data entry, bank cheque/draft processing, money counting machine, postal automation, etc. incorporate character recognition engines. Some of these applications work with handwritten characters also. Various classifiers have been deployed for such problems, of which the majority is as follows. Statistical methods extensively used include Naive Bayes Classifier, Bayesian Belief Network, Minimum Distance Classifier and K-Nearest Neighbor (K-NN). Artificial Neural Network (ANN) based classifiers include Multilayer Perceptron (MLP), Radial Basis Function (RBF), Probabilistic Neural Network, etc. Kernel based method is found in Support Vector Machines (SVM).

In this work, we only focus on most commonly used pattern classifiers in view of accuracy for three different datasets, which represent a wide range of handwritten data. It is ensured that all categories have been covered in our comparative study. The statistical classifiers considered for this work are Naive Bayes classifier, minimum distance classifier and K-NN. Among neural network based classifiers, MLP is chosen. Lastly, SVM is taken as a kernel based classifier.

Few works have been reported on performance comparison of classifiers. Arora et. al [1] have reported a comparative study of ANN and SVM for Devnagari character recognition. Zanaty et. al [2] have reported a similar study for data classification. Pal et. al [3], in their comparative study of Devnagari handwritten character recognition, have implemented fifteen different classifiers and showed that mirror image learning (MIL) performs the best. Mohiuddin et al. [4] have also carried out similar study on handprinted character recognition.

Other than character recognition, comparative study of classifiers for texture verification has been carried out by Chen et. al [5]. Comparative study of classifiers have also been applied for script recognition [6], stream data analysis [7], gesture recognition [8], depression identification from speech [9], glaucoma detection [10], breast cancer diagnosis [11], lipid layer classification [12], remote sensing [13], emotion detection [14], diabetes prediction [15] and network intrusion detection [16]. Barnaghi et. al [17] presented a

comparative study of only a few classifiers. They did not include SVM which is one of the most competent classifiers used today. While most comparative studies are carried out on real data, synthetic data has been used by Amancio et. al. [18].

However, comparative performance study for Arabic, Oriya and Bengali numerals is still left. Most of these studies stated above conjecture that MLP and SVM are most prospective classifiers. Considering some more statistical classifiers, this paper reports a comparative study for handwritten optical character recognition of Arabic, Oriya and Bengali numerals.

## 2. BRIEF OUTLINE OF CLASSIFIERS

Optical binarized samples of the dataset under consideration are normalized to a uniform size i.e. 48x48 pixels. Then, a total of 60 features viz. shadow (24), longest run (20) and octant centroid (16) [19] are extracted from each samples. Then, the feature vectors are directly passed to classifier for performance evaluation.

### 2.1 Naive Bayes Classifier

Naive Bayes Classifier predicts the class membership probabilities i.e. the probability that a given sample belongs to a particular class. It is based on Bayes theorem and assumes that the effect of an attribute value on a given class is independent of the values of other attributes. This assumption is called class conditional independence. It is made so as to simplify the computation involved and, in this sense, is considered 'Naive'. The naive Bayes classifier works as follows:

Let  $T$  be a training set of samples, each with their class labels. There are  $k$  classes i.e.  $C_1, C_2, \dots, C_k$ . Each sample is represented by an  $n$ -dimensional vector  $X = [x_1, x_2, \dots, x_n]$ , depicting  $n$  measured values of the  $n$  attributes  $A_1, A_2, \dots, A_n$  respectively. Given a sample  $X$ , the classifier will predict that  $X$  belongs to the class having the highest *a posteriori* probability conditioned on  $X$ . That is,  $X$  is predicted to belong to the class  $Y$  as  $Y = \operatorname{argmax}_{i \in \{1, \dots, k\}} P(C_i|X)$ .

By Bayes theorem, we get  $P(C_i|X) = P(X|C_i)P(C_i)/P(X)$ . As  $P(X)$  is the same for all classes, only  $P(X|C_i)P(C_i)$  needs to be maximized. If the class *a priori* probabilities  $P(C_i)$ , are not known, it is commonly assumed that the classes are equally likely, that is,  $P(C_1) = P(C_2) = \dots = P(C_k)$ , and we would therefore maximize  $P(X|C_i)$ . Assuming that the values of the attributes are conditionally independent of one another, given the class label of the sample,  $P(X|C_i)$  is computed as  $P(X|C_i) \cong \prod_{j=1}^n P(x_j|C_i)$ . Hence, the output class  $Y$  for  $X$  is given as  $Y = \operatorname{argmax}_{i \in \{1, \dots, k\}} P(C_i) \prod_{j=1}^n P(x_j|C_i)$ .

### 2.2 Minimum Distance Classifier

Let  $X = [x_1, x_2, \dots, x_n]$  be the feature vector for an unknown sample and  $C_1, C_2, \dots, C_k$  be the classes. Then, the error in matching  $X$  against a cluster centre  $\bar{X}_i$  is given by  $\|X - \bar{X}_i\|$  where  $\bar{X}_i$  denotes the mean of all samples of the class  $C_i$  and computed as  $\bar{X}_i = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]$ . Here,  $\|u\|$  is called the norm of the vector  $u$ . The distance is defined as an index of similarity so that the minimum error distance is identical to the maximum similarity. For finding the minimum error distance, we compute  $\|X - \bar{X}_i\|$  and choose the class for which this error is minimized.

### 2.3 k-Nearest Neighbor(s)

kNN algorithm is among the simplest of machine learning algorithms. The kNN is a type of instance based learning by relat-

ing unknown pattern to the known according to some distance. The k-nearest neighbor approach attempts to compute a classification function by examining the labeled training points as nodes or anchor points in the  $n$ -dimensional space, where  $n$  is the feature size. We calculate the Euclidean distance between the recall point and all the reference points in order to find  $k$  nearest neighbors, and then rank the obtained distances in ascending order and take the reference points corresponding to the  $k$  smallest Euclidean distances. The Euclidean distance  $D$  between an input feature vector  $X = [x_1, x_2, \dots, x_n]$  and a library feature vector  $C = [c_1, c_2, \dots, c_n]$  is given by Eq. 1.

$$D = \sqrt{\sum_{i=1}^n (c_i - x_i)^2} \quad (1)$$

where  $c_i$  is the  $i^{th}$  library feature and  $x_i$  is the  $i^{th}$  input feature and  $n$  is the number of features used for the classifier. Euclidean distance is the straight line distance between two points in  $n$ -dimensional space. A test sample  $X$  is then attributed the class label which corresponds to the majority of its  $k$  nearest (reference) neighbors.

### 2.4 Multilayer Perceptron

An artificial neuron combines its input signals  $a_i$  with corresponding weights  $w_i$  and a bias  $b$ . The output  $O$  is computed as  $O = f(b + \sum_{i=1}^n w_i a_i)$  where  $f$  is the activation function. A multi-layer perceptron is a network of such neurons arranged in layers. The first layer is the input layer and the last layer is the output layer. There are one or more intermediate layers. Consequently, the number of neurons at the input and output layers are the number of features and the number of classes respectively. The number of neurons at the intermediate layer(s) is determined empirically.

Learning a neural network is adjusting the weights for the neurons from the training samples. Back-propagation is the most widely used supervised learning algorithm. It is also called as generalized delta algorithm because it extends the means to train a one layer perceptron (delta rule). It is based on minimizing the difference between the desired output and the obtained output, through descent error gradient method.

In a back-propagation neural network, the learning algorithm has two phases. First, a training input pattern is presented to the network input layer. The network then propagates the input pattern from layer to layer until the output pattern is generated by the output layer. If this pattern is different from the desired output, an error is calculated and then propagated backward through the network from the output layer to the input layer. The weights are modified as the error is propagated. Like any other neural networks, a back-propagation one is determined by the connections between neurons, the activation function used by the neurons and the learning algorithm that specifies the procedure for adjusting weights.

### 2.5 Support Vector Machines

Support vector machines use a nonlinear mapping to transform the original training data into a higher dimension. It searches for the linear optimal separating hyper-plane with an appropriate number of support vectors (SVs). Data from two classes can always be separated by a hyper-plane. The SVM finds this hyper-plane using support vectors and margins.

Though, the training time of even the fastest SVMs can be extremely high, they are highly accurate owing to their ability to model complex nonlinear decision boundaries. They are much less prone to over-fitting compared to other methods.



Fig. 1. Some handwritten samples of Arabic numerals (Please note that the samples of the same class are extensively varying in shape and appearance)

An SVM is basically a binary classifier with discriminate function being the weighted combination of kernel functions over all training samples. The weights (coefficients) are learned by quadratic programming with the aim of maximizing the margin in feature space. After learning, the samples of non-zero weights are called support vectors (SVs), which are stored and used in classification. When samples are linearly separable, we have a hyper-plane  $wx + b = 0$  that separates all the training samples as  $x_i \cdot w + b > 1$  if  $y_i = +1$  and  $x_i \cdot w + b \leq 1$  if  $y_i = -1$  where  $w$  is normal to the hyper-plane.

For linearly inseparable scenario, various kernels are used to transform the samples so as to make them linearly separable. For a multi-class SVM, there can be three approaches to pattern classification viz. one against one, one against all and DAG SVM. In this work, we have incorporated one against all approach because of its effectiveness [20].

### 3. EXPERIMENTAL RESULTS

The classifiers discussed above have been deployed on three datasets viz. Arabic, Bengali and Oriya numerals. Necessary tuning is done separately for each dataset. For the MLP, only one hidden layer is chosen to keep the computational requirement of the same low without affecting its function approximation capability. Several runs of back propagation algorithm with different learning rate are executed for different numbers of neurons in its hidden layer.

#### 3.1 Performance on Arabic Numerals Dataset

Arabic numerals dataset developed by CMATER Research Lab of Jadavpur University, Kolkata, India has been chosen for the present experiment. It contains 3000 isolated image samples of handwritten Arabic numerals (300 samples for each numeral). Some samples of this dataset are shown in Fig. 1. This dataset is randomly divided into training set and test set with 2:1 ratio.

The K-NN classifier requires to specify the number of nearest neighbour samples for classification. The recognition performance of the K-NN, for the test samples with different number of nearest neighbor is shown in Fig. 2. It may be noted that when  $k = 3$ , the classifier gives the best performance and when the value of  $k$  is increased, the performance degrades.

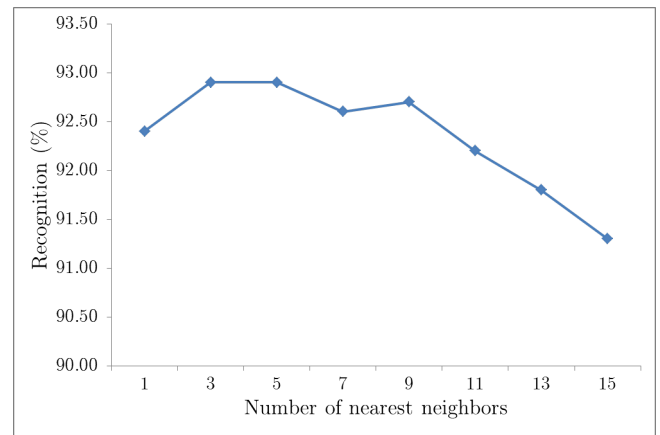


Fig. 2. Recognition performances of the K-NN with different numbers of nearest neighbors for Arabic dataset

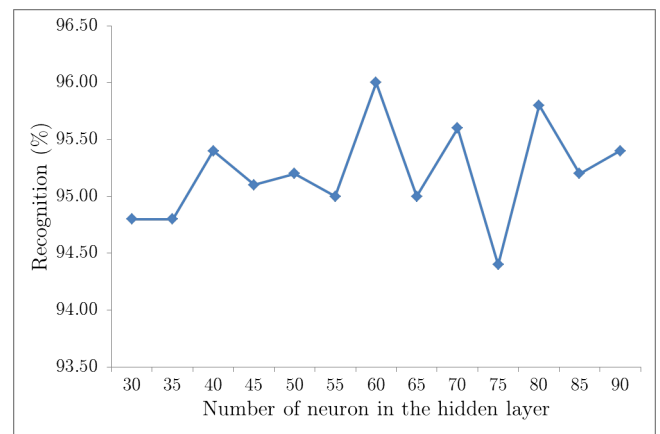


Fig. 3. Recognition performances of MLP with different numbers of neurons in the hidden layers for the Arabic Numerals (Learning rate  $\eta = 0.15$ )

Recognition performance of the MLP on the Arabic test samples for different number of neurons in the hidden layer is shown in Fig. 3. It may be observed that the highest performance is achieved when the number of neuron at the hidden layer is 60.

Following one versus one approach of SVM,  $^{10}C_2 = 45$  binary classifiers are constructed. Among them, the class for which the highest vote is achieved is considered as the class of the recall pattern. We also set parameter  $\sigma = 10$  (Gaussian RBF kernel) and  $d = 3$  for polynomial kernel function heuristically. Table 1 shows the classification accuracy obtained with all classifiers.

#### 3.2 Results on Oriya Numerals Dataset

A dataset of 6000 isolated handwritten Oriya numerals, provided by CVPR Unit, Indian Statistical Institute, Kolkata has been chosen for performance comparison on Oriya numerals. The ratio of training and test samples is same as that of Arabic dataset. Some sample images from this dataset is shown in Fig. 4.

kNN yields the highest performance i.e. 91.50% when  $k = 1$ . MLP gives the highest performance of 95.20% when number of hidden layer neuron is 65. Other tuning is done as mentioned in Section 3.1. Recognition rates for other classifiers are shown in Table 2.

Table 1. Recognition rate of various classifiers on Arabic test samples

kNN	Naive Bayes	Minimum Distance	SVM Linear	SVM Quadratic	SVM Polynomial	SVM RBF	MLP
92.90	81.10	73.50	95.20	94.20	95.20	96.70	96.00

Table 2. Recognition rate of various classifiers on Oriya test samples

kNN	Naive Bayes	Minimum Distance	SVM Linear	SVM Quadratic	SVM Polynomial	SVM RBF	MLP
91.50	85.50	78.10	94.20	92.90	90.00	95.10	95.20



Fig. 4. Some handwritten samples of Oriya numerals (It may be noted that the samples of the same class are extensively varying in shape and appearance)

### 3.3 Results on Bengali Numerals Dataset

For the present work, experiments have been carried out on a dataset of 3000 isolated handwritten Bengali numerals (300 samples for 10 numerals), provided by CMATER Research Lab of Jadavpur University, Kolkata, India. Similar to previous datasets, the ratio of training and test samples is taken as 2:1. Some samples from the said dataset is shown in Fig. 5.

Similar to the previous datasets, parameter tuning is done. It is found that when  $k = 1$ , the K-NN classifier gives the optimal performance i.e. 90.80%. The MLP gives the highest performance i.e. 95.10% when the number of neurones at the hidden layer is 40. Table 3 shows the the recognition rates achieved for all classifiers.

### 3.4 Discussion

A summary of classification accuracies obtained for three datasets is shown in Fig. 6. It shows that MLP yields the highest performance in most cases. It gives recognition rate of 96.00%, 95.20% and 95.10% for Arabic, Oriya and Bengali numerals respectively. In MLP, number of neurons at the hidden layer is chosen experimentally for every datasets. However, RBF kernel produces better classification i.e. 96.70% for Arabic numerals.

Selection of feature sets, feature optimization, post-processing and/or pre-processing can significantly contribute to the classification accuracy for all classifiers. But, these are not performed as the objective is to compare the performance of classifiers. Hence, a uniform platform, an essential requirement for the same is adopted in the present work.



Fig. 5. Some handwritten samples of Bengali isolated numerals (Extensive variability in shape and appearance for the samples of the same class may be noted)

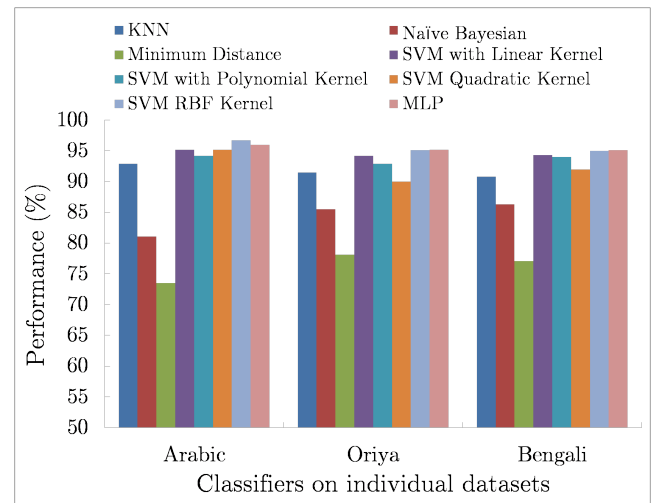


Fig. 6. Overall performances of classifiers on all datasets (Arabic, Oriya and Bengali numerals dataset)

It may be noted that when the degree of polynomial kernel is increased, over-fitting may take place. As a result, the classification performance has degraded in Oriya as well as Bengali datasets with the increase in the degree of the polynomial. For Arabic datasets, when the degree of polynomial kernel is 4, it performs very poor. However, RBF kernel performs better compared to all polynomial kernels.

Table 3. Recognition rate of various classifiers on Bengali test samples

kNN	Naive Bayes	Minimum Distance	SVM Linear	SVM Quadratic	SVM Polynomial	SVM RBF	MLP
90.80	86.30	77.10	94.30	94.00	92.00	95.00	95.10

It may also be noted that compared to other classifiers, MLP performs well for noisy patterns. As handwritten samples vary extensively, MLP yields the best performance in most cases.

#### 4. CONCLUSION

In this paper, a comparative performance study of various statistical, ANN and kernel based classifiers for handwritten numerals recognition of three scripts namely Arabic, Oriya and Bengali is reported. It is found that Multilayer Perceptron performed the best for Oriya (95.20%) and Bengali (95.10%) datasets, and SVM with RBF kernel performs the best for Arabic (96.70%) dataset. MLP performs the second highest (96.00%), and SVM linear and polynomial ( $d = 3$ ) kernels jointly performs the third highest for Arabic dataset.

For the Oriya dataset, SVM with RBF kernel gives the second highest (95.10%) and SVM linear kernel gives the third highest (94.20%) performance. In Bengali numerals dataset also, it is found that SVM RBF kernel is second highest (95.00%) and linear kernel (94.30%) is the third highest in performance. Among other classifiers, kNN is giving relatively better results.

In all cases, minimum distance classifier gives the worst performance. Although, post-processing of the classified patterns may improve classification performance, it is not performed due to the objective of their comparison. However, this experiment can be further carried out for alphabets as well and the comparison can be made with respect to the other feature sets.

**Acknowledgement.** Authors are thankful to CMATER Research Lab of Jadavpur University for providing Arabic and Bengali datasets and CVPR Unit of Indian Statistical Institute for providing Oriya numerals dataset. They are also thankful to Department of Computer Science and Engineering, Aliah University for providing infrastructural support for carrying out this work.

#### 5. REFERENCES

- [1] S. Arora, D. Bhattacharjee, M. Nasipuri, L. Malik, M. Kundu and D. K. Basu, "Performance Comparison of SVM and ANN for Handwritten Devnagari Character Recognition", *Int. J. of Computer Science Issues*, vol. 7, no. 3, pp. 18-26.
- [2] E. A. Zanaty, "Support Vector Machines (SVMs) versus Multilayer Perception (MLP) in data classification", *Egyptian Informatics Journal* (2012) 13, 177183.
- [3] U. Pal, T. Wakabayashi, F. Kimura, "Comparative Study of Devnagari Handwritten Character Recognition using Different Feature and Classifiers", *Proc. 10th International Conference on Document Analysis and Recognition*, pp. 1111-1115, 2009.
- [4] K. Mohiuddin and Jianchang Mao. "A comparative study of different classifiers for handprinted character recognition", *Pattern Recognition in Practice IV* (2014): 437-448.
- [5] D. Chen and J. M. Odobez "Comparison of Support Vector Machine and Neural Network for Text Texture Verification", *IDIAP*, 2002.
- [6] P. K. Singh, R. Sarkar, N. Das, S. Basu and M. Nasipuri, "Statistical comparison of classifiers for script identification from multi-script handwritten documents". *International Journal of Applied Pattern Recognition*, 1(2), pp.152-172, 2014.
- [7] A. Patil and V. Attar, "Framework for performance comparison of classifiers", In *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011)*, December 20-22, pp. 681-689, Springer India, 2012.
- [8] H. H. Avils-Arriaga, L. E. Sucar-Succar, C. E. Mendoza-Durn and L. A. Pineda-Corts, "A comparison of dynamic naive bayesian classifiers and hidden markov models for gesture recognition", *Journal of Applied Research and Technology*, 9(1), pp. 81-102, 2011.
- [9] S. Alghowinem, R. Goecke, M. Wagner, J. Epps, T. Gedeon, M. Breakspear and G. Parker, "A comparative study of different classifiers for detecting depression from spontaneous speech", *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8022-8026, 2013.
- [10] W. Adler, A. Peters and B. Lausen, "Comparison of classifiers applied to confocal scanning laser ophthalmoscopy data", *Methods of Information in Medicine*, 47(1), 38-46, 2008.
- [11] G. I. Salama, M. B. Abdelhalim and M. A. E. Zeid, "Experimental comparison of classifiers for breast cancer diagnosis", *Proc. of Seventh International Conference on Computer Engineering and Systems (ICCES)*, pp. 180-185, IEEE, 2012.
- [12] B. Remeseiro, M. Penas, A. Mosquera, J. Novo, M. G. Penedo and E. Yebra-Pimentel, "Statistical comparison of classifiers applied to the interferential tear film lipid layer automatic classification", *Computational and Mathematical Methods in Medicine*, 2012.
- [13] R. S. Dwivedi, S. Kandrika, and K. V. Ramana, "Comparison of classifiers of remote-sensing data for land-use/land-cover mapping", *Current Science*, 86(2), 328-334, 2004.
- [14] I. Shafran and M. Mohri, "A Comparison of Classifiers for Detecting Emotion from Speech", *Proc. of ICASSP*, pp. 341-344, 2005.
- [15] N. Nai-arun, and R. Moungrai, "Comparison of Classifiers for the Risk of Diabetes Prediction", *Procedia Computer Science*, 69, 132-142, 2015.
- [16] P. Aggarwal and S. K. Sharma, "An empirical comparison of classifiers to analyze intrusion detection", In *Fifth International Conference on Advanced Computing and Communication Technologies*, pp. 446-450, IEEE, 2015.
- [17] P. M. Barnaghi, V. A. Sahzabi, A. A. Bakar, "A comparative study for various methods of classification", In *International Conference on Information and Computer Networks*, vol. 27, no. 2, pp. 875-81, 2012.
- [18] D. R. Amancio, C. H. Comin, D. Casanova, G. Travieso, O. M. Bruno, F. A. Rodrigues, L. da Fontoura Costa, "A systematic comparison of supervised classifiers", *PloS one*, 9(4), e94137, 2014.
- [19] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, D. K. Basu, "Handwritten Bangla Alphabet recognition using an MLP based classifier", *NCCPB-2005, Bangladesh*, pp.285-291.
- [20] Ben Aisen "A Comparison of Multiclass SVM Methods", December 15, 2006.