CHAPTER 7

RECOGNITION OF OFFLINE HANDWRITTEN GURUMUKHI WORDS USING EXTREME GRADIENT BOOSTING (XGBOOST) TECHNIQUE

In this chapter, the holistic approach to offline handwritten Gurumukhi word recognition system based on eXtreme Gradient Boosting (XGBoost) has expatiated. XGBoost technique is employed to boost the system performance due to its higher efficiency as compared to other algorithms. For the present work, the efficiency of the XGBoost technique has been explored. To extract significant features from the word images, four feature extraction techniques, namely, zoning features, diagonal features, intersection & open-end points features, and peak extent features (horizontally and vertically) have been considered. This whole chapter is segregated into 5 sections. Section 7.1 elaborates the concept of the XGBoost technique. To assess the system performance, several evaluation measures have been considered which are discussed in section 7.2. Section 7.3 discusses the experimental results based on the considered features and the XGBoost technique. The comparative analysis of the present work is demonstrated in section 7.4. Finally, the complete chapter is summarized in section 7.5.

7.1 XGBOOST TECHNIQUE [CHEN AND GUESTRIN, 2016]

XGBoost or eXtreme Gradient Boosting is an ensemble machine learning technique that consists of a sequence of Decision Trees. It is called gradient boosting which is employed to boost the weak learners/classifiers and build a predicted model by the integration of weak classifiers. This algorithm works by assigning similar weights to all the training specimens, and this refers to the probability by which the record gets selected by the classifier for the purpose of training. Similar weights specify the equal probability of selection of all the records. After getting trained, the model makes a prediction. The classifier that incorrectly classifies the records is called the weak classifier and then the weights are updated to reduce the errors of the existing model which are then inputted to the second classifier. The second classifier chooses those records for training which are having maximum weights. Therefore, weight updation plays a key role in the XGBoost technique. This process goes on for Decision Trees

one after another sequentially up to the *n*th Decision Tree. Each of the weak classifiers generates some prediction and the final prediction of the test sample is made on the basis of a maximum of identical predictions made by the weak classifiers as depicted in Figure 7.1.

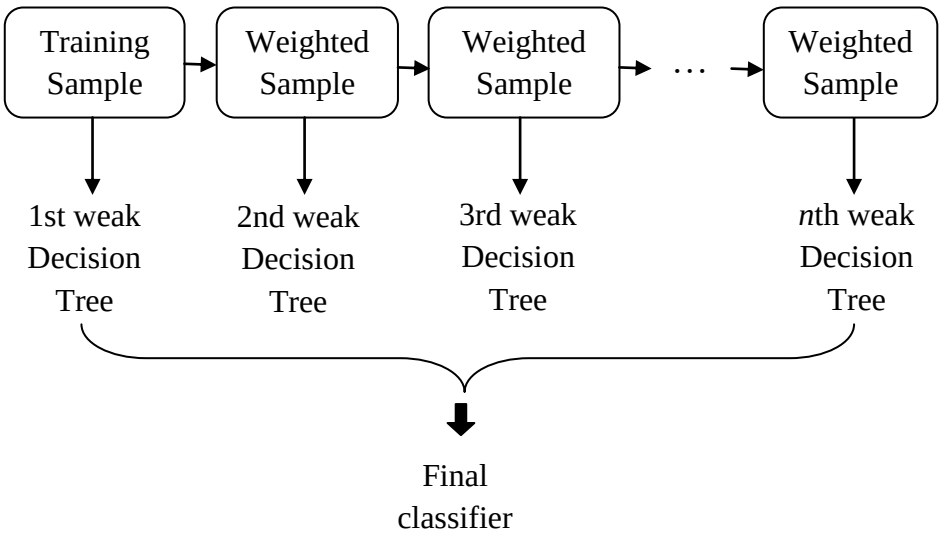

Figure 7.1. Working of XGBoost technique

7.1.1 Features of XGBoost Technique

- It is easy to utilize this algorithm and it provides better efficiency and higher accuracy in comparison to other algorithms.
- It allows the user to carry out cross-validation at each iteration of the boosting process; and thus, acquires the accurate optimum number of iterations in a unique run.
- It aids in parallel processing and possesses higher speed as compared to Gradient Boosting Machine (GBM).
- It works well even with missing values.
- To sort out the over-fitting issue, it has a built-in L1 and L2 regularization facility. Due to this facility, it is also termed as a regularized form of GBM.

7.2 EVALUATION MEASURES

To measure the system performance, several evaluation measures such as CPU elapsed time, Accuracy, Precision, Recall, F1-Score and Area Under Curve (AUC) have been utilized. Accuracy has been explained in chapter 4, whereas Precision, Recall and F1-Score have already been explained in chapter 5. The rest of the evaluation measures are explained in the following sub-sections:

7.2.1 CPU elapsed time

To assess the processor speed, the CPU elapsed time is utilized that is observed as inversely proportional to the execution time. CPU elapsed time is stated in milliseconds (ms).

7.2.2 Area Under Curve (AUC)

It demonstrates the possibility that the randomly selected positive specimen will be ranked higher by the classifier as compared to the randomly selected negative specimen. It is utilized for binary classification. It is computed based on ROC (Receiver Operating Characteristics) curve by plotting FPR (False Positive Rate) along the x-axis and TPR (True Positive Rate) along the y-axis as depicted in Figure 7.2. It lies between 0 and 1, where the higher value corresponds to the superior performance of the system.

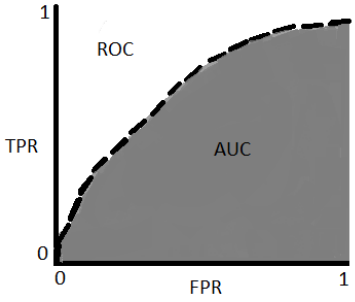

Figure 7.2. Area Under Curve(AUC)

7.3 EXPERIMENTAL RESULTS

In order to perform the experiments, a public benchmark dataset was considered that comprises 40,000 word samples corresponding to 100 distinct place names gathered from 40 distinct writers. This dataset is publicly available at https://sites.google.com/

view/gurmukhi-benchmark/home/word-level-gurmukhi-dataset (Kaur and Kumar, 2019) for researchers. The dataset is partitioned into training and testing set based on three partitioning strategies as delineated in Table 7.1. In the first strategy, the complete dataset has been partitioned into 90% training and 10% testing set. Whereas in strategy b, 80% of data has been considered as a training set and the remaining 20% data as the testing set. In the last strategy c, 70% and 30% data has been taken in training and testing set, respectively.

Table 7.1. Dataset Partitioning Strategies

Partitioning Strategy	Words (Training Set)	Words (Testing Set)
a	36,000 (90%)	4,000 (10%)
b	32,000 (80%)	8,000 (20%)
c	28,000 (70%)	12,000 (30%)

To evaluate the system, 85 zoning features, 85 diagonal features, 170 intersection & open-end points features, and 170 peak extent features were extracted as discussed in chapter 4 and chapter 6. All these features were extracted from the complete word image without segmenting the word into its constituent characters. The system performance was assessed based on the considered features and XGBoost technique using six considered evaluation measures as discussed in the following sub-sections.

7.3.1 System performance based on zoning features

Employing the zoning features to the XGBoost technique, the best system performance was achieved as 91.66% (accuracy), 91.39% (precision), 91.66% (recall), 91.14% (F1-score), and 95.66% (AUC) based on 90:10 partitioning strategy as delineated in Table 7.2. The present system obtained the best CPU elapsed time of 43.63 ms using 80% training and 20% testing set.

Table 7.2. System evaluation using zoning features

Zoning Features	**Evaluation Measures**					
Partitioning strategy (Training: Testing)	**CPU Elapsed Time**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
90:10	50.09 ms	91.66%	91.39%	91.66%	91.14%	95.66%
80:20	43.63 ms	91.34%	91.21%	91.34%	91.04%	95.50%
70:30	46.33 ms	89.93%	90.01%	89.93%	89.84%	94.76%

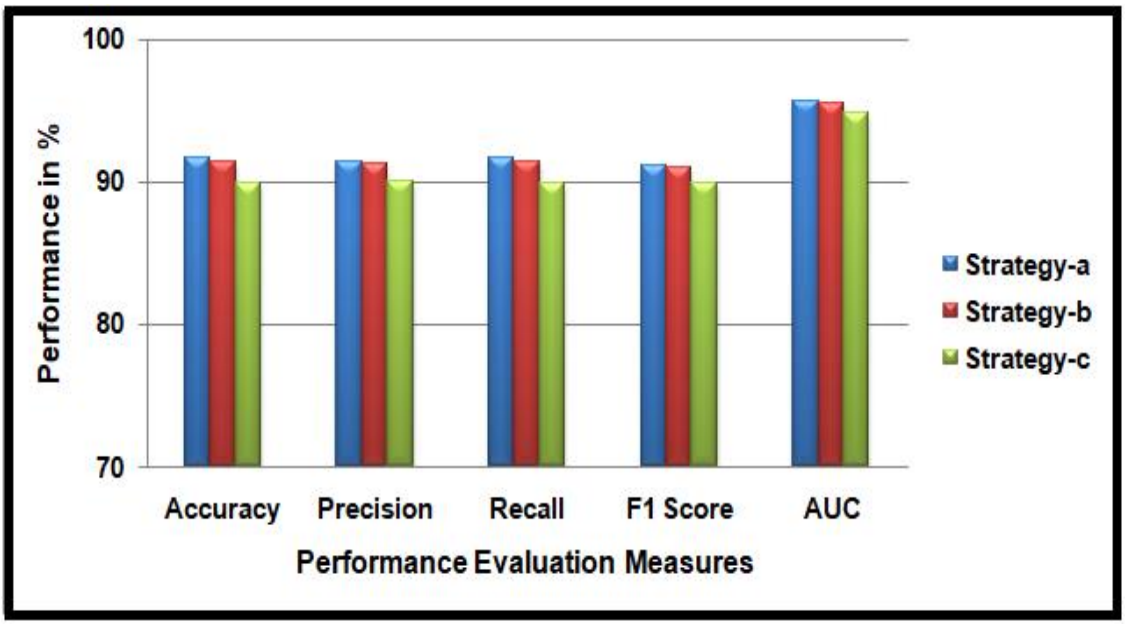

Figure 7.3. System evaluation using zoning features

7.3.2 System performance based on diagonal features

Based on diagonal features, the XGBoost technique achieved the best system performance as 91.30% (accuracy), 91.30% (recall), and 95.47% (AUC) by utilizing a 90:10 partitioning strategy, whereas precision and F1-score of 91.03% and 90.88% were attained by utilizing 80:20 partitioning strategy as delineated in Table 7.3. The system achieved the best CPU elapsed time of 42.77 ms based on 70% training and 30% testing set.

Table 7.3. System evaluation using diagonal features

Diagonal Features	**Evaluation Measures**					
Partitioning strategy (Training: Testing)	**CPU Elapsed Time**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
90:10	51.60 ms	91.30%	90.95%	91.30%	90.73%	95.47%
80:20	46.86 ms	91.18%	91.03%	91.18%	90.88%	95.41%
70:30	42.77 ms	90.00%	90.06%	90.00%	89.92%	94.80%

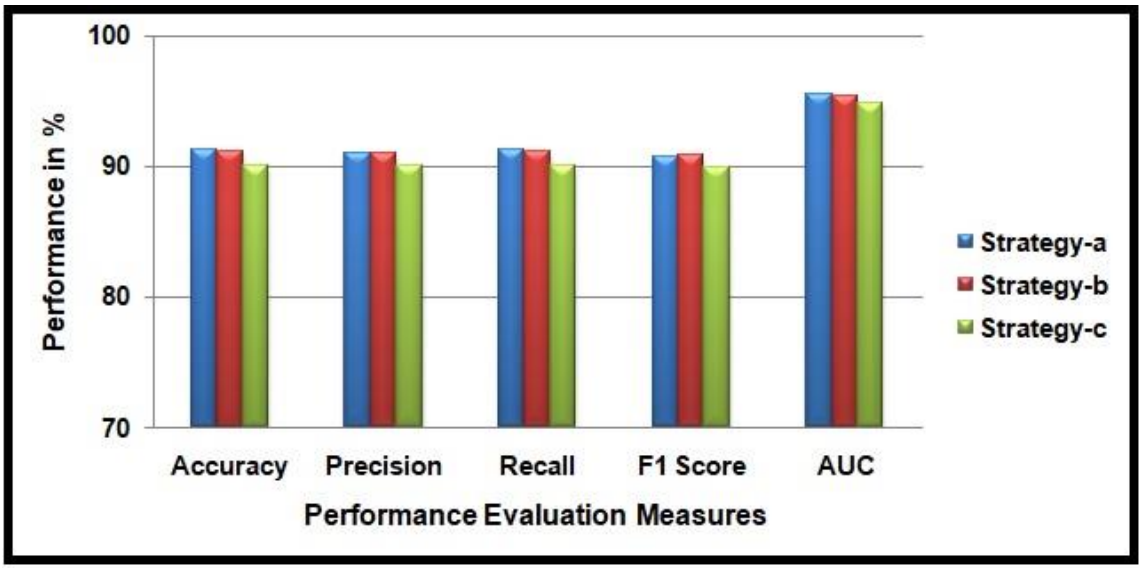

Figure 7.4. System evaluation using diagonal features

7.3.3 System performance based on intersection & open-end points features

Considering the intersection & open-end points features to XGBoost technique, the system attained best performance as 88.37% (accuracy), 88.40% (precision), 88.37% (recall), and 93.94% (AUC) based on 90:10 partitioning strategy, whereas the system obtained best F1-score of 87.95% using 80:20 partitioning strategy as delineated in Table 7.4. The system attained the best CPU elapsed time of 57.20 ms based on a 70:30 partitioning strategy.

Table 7.4. System evaluation using intersection & open-end points features

Intersection & Open-end Points	**Evaluation Measures**					
Partitioning strategy (Training: Testing)	**CPU Elapsed Time**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
90:10	73.51 ms	88.37%	88.40%	88.37%	87.87%	93.94%
80:20	67.28 ms	88.31%	88.24%	88.31%	87.95%	93.91%
70:30	57.20 ms	87.22%	87.54%	87.22%	87.21%	93.35%

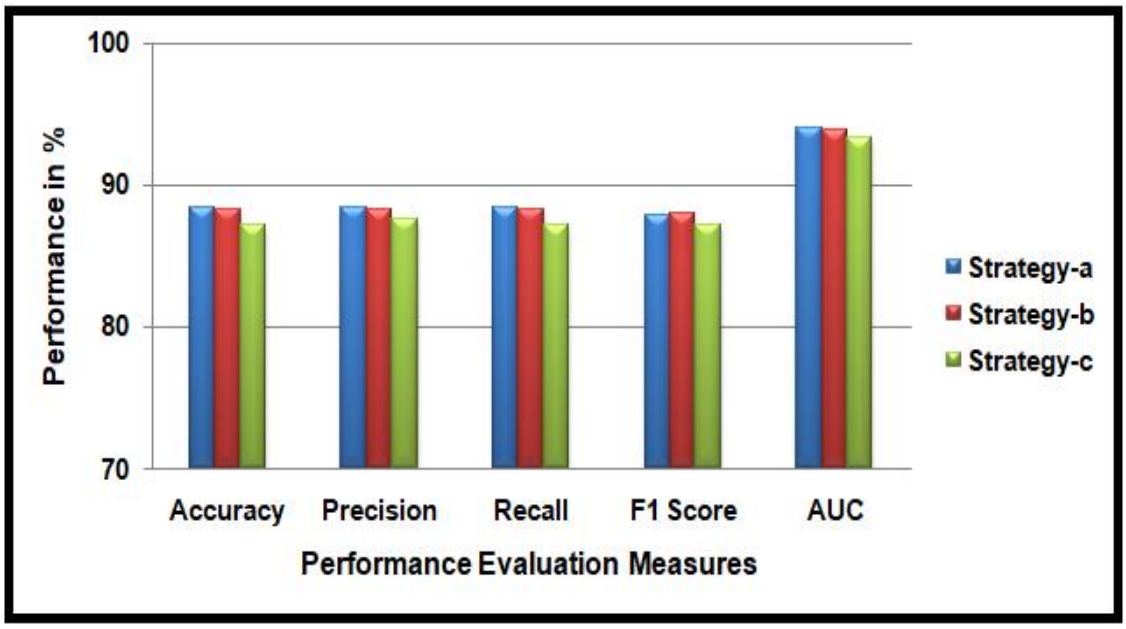

Figure 7.5. System evaluation using intersection & open-end points features

7.3.4 System performance based on peak extent features

By employing peak extent features to XGBoost technique, the system had the best performance as 86.27% (accuracy), 86.13% (precision), 86.27% (recall), 85.73% (F1-score), and 92.85% (AUC) based on 90:10 partitioning strategy as delineated in Table 7.5. The system gained the best CPU elapsed time of 57.23 ms based on 70% training and 30% testing dataset.

Table 7.5. System evaluation using peak extent features

Peak Extent Features	**Evaluation measures**					
Partitioning strategy (Training: Testing)	**CPU Elapsed Time**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
90:10	79.16 ms	86.27%	86.13%	86.27%	85.73%	92.85%
80:20	64.82 ms	84.44%	84.53%	84.44%	84.26%	91.90%
70:30	57.23 ms	83.36%	83.27%	83.36%	83.12%	91.34%

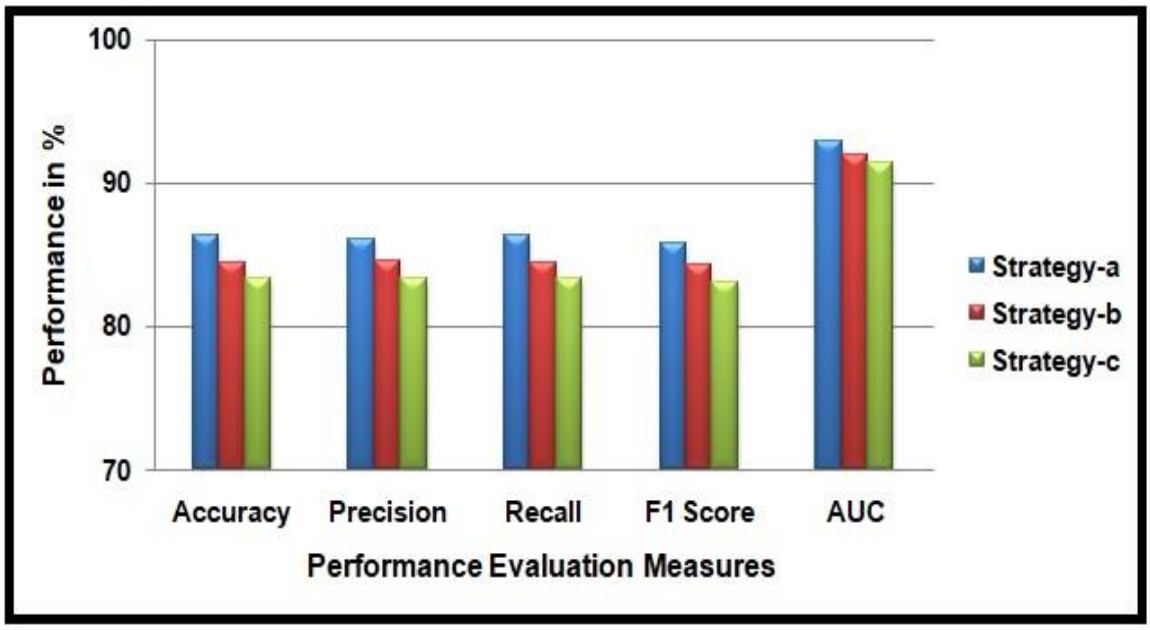

Figure 7.6. System evaluation using peak extent features

7.4 COMPARISON WITH THE EXISTING APPROACHES AND SYNTACTIC ANALYSIS

In the literature, there is the existence of several machine learning approaches such as Hidden Markov Model (HMM), k-Nearest Neighbor (k-NN), Gabor filters with Support Vector Machine (SVM), Multilayer Perceptron (MLP), and several feature selection approaches like Genetic Algorithm (GA), Memetic Algorithm (MA), Harmony Search (HS) and Particle Swarm Optimization (PSO) in order to recognize

handwritten words in different scripts. The comparative analysis of the present approach with these state-of-the-art approaches has been provided by us as delineated in Table 7.6. The present approach has also been compared with Gurumukhi character recognition approaches as outlined in Table 7.7.

Table 7.6. Comparison of the present approach with state-of-the-art approaches

Authors	**Script**	**Dataset**	**Feature Extraction/ Selection Technique**	**Classification Technique**	**Accuracy**
Kessentini *et al.* (2010)	Arabic and Latin	(i) IFN/ENIT (ii) IRONOFF	Density and contour based features	HMM	(i) 79.8% (ii) 89.8%
Patel *et al.* (2015b)	Latin	300 handwritten English words	Structural features	k-NN	90%
Das *et al.* (2016)	Bangla	1020 handwritten words	Harmony Search (HS) feature selection approach	MLP	90.29%
Assayony and Mahmoud (2017)	Arabic	CENPARMI	Gabor filters fused with Bag-of-features	SVM	86.44%
Tavoli *et al.* (2018)	Arabic	(i) Iran-cities (ii) IFN/ENIT (iii) IBN SINA	Statistical Geometric Components of Straight lines(SGCSL)	SVM	(i) 67.47% (ii) 80.78% (iii) 86.22%

Authors	**Script**	**Dataset**	**Feature Extraction/ Selection Technique**	**Classification Technique**	**Accuracy**
Arani *et al.* (2019)	Farsi	Iranshahr 3	Image gradient, black-white transitions and contour chain code features	HMM and MLP	89.06%
Ghosh *et al.* (2019)	Bangla	7500 handwritten words	Gradient -based features and modified Statistical and Contour based features; MA based wrapper filter selection approach	MLP	89.67% (without feature selection) 93% (with feature selection)
Present Approach	Gurumukhi	40,000 handwritten words	(i) Zoning features (ii) Diagonal features (iii) Intersection & open-end points features (iv) Peak extent features	XGBoost	(i) 91.66% (ii) 91.30% (iii) 88.37% (iv) 86.27%

Table 7.7. Comparison of the present approach with state-of-the-art Gurumukhi character recognition approaches

Authors	**Dataset (Character Specimens)**	**Feature Extraction/ Selection Technique**	**Classification Technique**	**Accuracy**
Kumar *et al.* (2013b)	7000	Centroid, horizontal peak extent, vertical peak extent, shadow features	(i) Linear-SVM (ii) k-NN (iii) MLP	(i) 95.62% (ii) 95.48% (iii) 94.74%
Kumar *et al.* (2014a)	3500	Centroid, diagonal, horizontal peak extent, vertical peak extent features; Correlation-based feature selection (CFS), Principal Component Analysis (PCA) and Consistency-Based (CON) feature selection	SVM	91.80% (PCA)
Kumar *et al.* (2014b)	3500	Parabola curve fitting and power curve fitting based features	(i) SVM (ii) k-NN	(i) 97.14% (ii) 98.10%
Kumar *et al.* (2016)	7000	Boundary extent feature extraction; PCA	(i) k-NN (ii) SVM (iii) MLP	93.8% (RBF-SVM)
Kumar *et al.* (2017)	10,500	Discrete cosine transformations, discrete wavelet transformations, fast Fourier transformations and fan beam transformations	SVM	95.8% (Discrete cosine transformations)

Authors	**Dataset (Character Specimens)**	**Feature Extraction/ Selection Technique**	**Classification Technique**	**Accuracy**
Present Approach	40,000 Gurumukhi handwritten words	(i) Zoning features (ii) Diagonal features (iii) Intersection & open-end points features (iv) Peak extent features	XGBoost	(i) 91.66% (ii) 91.30% (iii) 88.37% (iv) 86.27%

After a comparative analysis of the present approach with state-of-the-art approaches, the following key points have been analyzed:

- The present approach is the first attempt of its type as the XGBoost technique has not been utilized for the recognition of offline handwritten Gurumukhi words before.
- By performing experiments on a public benchmark dataset (Kaur and Kumar, 2019) of Gurumukhi script, the XGBoost technique attained the maximum word recognition accuracy of 91.66% based on zoning features which are found to be superior as compared to the accuracies attained via some existing approaches of other scripts (Kessentini *et al.*, 2010; Patel *et al.*, 2015b; Assayony and Mahmoud, 2017; Tavoli *et al.*, 2018; Arani *et al.*, 2019).
- XGBoost technique achieved better recognition accuracy than the approach presented by Ghosh *et al.* (2019) where the latter approach was based on the original feature set without any feature selection technique. But with the incorporation of the feature selection technique, the recognition accuracy of the latter approach exceeded that of the present approach.
- The present approach surpassed the HS based feature selection technique proposed by Das *et al.* (2016), wherein the latter approach, the dimensionality of the feature set comprising 65 elliptical features got reduced to 48 features with enhancement in word recognition accuracy from 81.37% to 90.29% for Bangla script.

- Even without incorporating any feature selection technique, the recognition accuracy achieved by the present approach is very close to the rate attained via Kumar *et al.* (2014a). Thus, the present approach's accuracy can further be enhanced by incorporating feature selection techniques.

7.5 CHAPTER SUMMARY

This chapter elaborates the offline handwritten word recognition system in Gurumukhi script based on a holistic approach that goes for the recognition of complete word without any segmentation process. To extract the desirable attributes from the words (place names), four feature extraction approaches such as zoning features, diagonal features, intersection & open-end points features, and peak extent features were incorporated. These extracted features were then given as an input to the XGBoost technique to classify the considered word into one of the 100 different places. The system achieved the best performance in the case of zoning features as 91.66% (accuracy), 91.66% (recall), 91.39% (precision), 91.14% (F1-score), and 95.66% (AUC) on the basis of a 90:10 partitioning strategy, where 36,000 words were used to train the model and remaining 4,000 words were used to test the model. It has also been perceived that the system attained minimum CPU elapsed time based on a 70:30 partitioning strategy for all the considered features except zoning features. The comparative analysis reveals that the XGBoost technique proves to be a better machine learning model in order to recognize offline handwritten Gurumukhi words.