



Improving the Accuracy of Handwritten Digits Recognition through Parameter Tuning of XGBoost Algorithm

C. Beulah Christalin Latha¹, Sujni Pau² and S. Carolin Jeeva³

¹Assistant Professor, Department of Digital Sciences,
Karunya Institute of Technology and Sciences, Coimbatore, (Tamilnadu), India.

²Faculty, Higher Colleges of Technology, Dubai, United Arab Emirates.

³Assistant Professor, Department of Digital Sciences,
Karunya Institute of Technology and Sciences, Coimbatore, (Tamilnadu), India.

(Corresponding author: C. Beulah Christalin Latha)

(Received 01 May 2020, Revised 22 June 2020, Accepted 01 July 2020)

(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Recognition of handwritten characters or digits has gained attention in the research field in the recent past in the areas of machine learning, pattern recognition and computer vision. Recognition of digits and characters play a crucial role in educating learners through machine interfaces and they are even used in educating differently abled people. Recognition of handwritten characters is a challenging task because the writing style is different for different writers. The patterns change for the same person also when the writing speed or pen differs. Similarity between the digits also makes the identification process more challenging. Broken digits, incomplete digits and digits with extra strokes also makes the process more difficult. Classification is a machine learning technique that is used to predict unknown data based on a model that is built by training known data. This paper uses multiple classification algorithms for identifying handwritten digits and analyzes their performance. Individual classifiers as well as ensemble of classifiers have been used in this study. It has been inferred that the extreme gradient boosting classifier outperforms the other individual classifiers and also other ensemble classifiers in terms of accuracy of prediction of handwritten digits. The performance of the algorithm is further improved by tuning the hyper parameters. The proposed approach improves the accuracy of recognition of handwritten digits and identifies them with an accuracy of 99.17%. This is a machine learning approach for the identification of handwritten digits with the highest accuracy.

Keywords: Handwritten digit recognition, machine interfaces for education, Ensemble classification, Performance Analysis of Classifiers, Extreme gradient boosting.

Abbreviations: SVM, Support Vector Machines; K-NN, k-Nearest Neighbor; XGBoost, Extreme Gradient Boosting; HMM, Hidden Markov Models; CNN, Convolutional Neural Networks; MLP, Multilayer Perceptron; PCA, Principal Component Analysis.

I. INTRODUCTION

Recognition of handwritten characters and digits is a vital task in the field of computer vision. Automatic recognition of handwritten images and characters are essential when handwritten documents need to be processed automatically using computers and other machines. Many documents written during olden days are available only in handwritten form. Handwritten character recognition may also be used in the field of education for evaluating answer scripts. Such systems may also find their applications in detecting characters in paper documents, pictures and touch screen devices. Recognition of handwritten characters is considered to be a critical issue since there are deep variation in individual's writing styles, shape and scale of the digits, context of the digits, different writing devices and media. Image processing using mathematical morphology is one of the traditional techniques used for handwritten digit recognition. Such methods are usually complex and time consuming since they have to detect horizontal and vertical lines and concavities to identify the characters. Problems were identified while detecting

broken digits, incomplete digits and digits with extra strokes. Some of these systems were unable to identify all the digits [8, 12]. Kumar *et al.*, (2010) proposed an improved method by using a morphological classification tree for recognizing handwritten digits. In this method, the digits were classified into two groups: one group consists of blobs with/without stems and the other digits with stem only. The method recognizes all the ten digits by using the morphological operators such as dilation, thinning, region filling and connected approach to extract various features and check for topological configurations. An accuracy of 90% was obtained in this method [21]. The researches not only include Arabian numerals but also focus on Chinese characters, Arabic words and Persian numerals [1, 4, 8, 12].

Machine learning techniques are used in the recent past for recognizing handwritten characters and digits. Neural networks are one of the common techniques used for handwritten digit identification [1, 6, 17, 18, 20, 23]. A deep convolutional neural network has been used for the recognition of Arabic Handwritten digits. An

accuracy of 95.7% was achieved in this technique [7]. Álvarez *et al.*, (1998) achieved an accuracy of 96.2% multilayer and clustered neural networks trained with the back propagation algorithm for recognition of handwritten and printed digits [5]. Abdelhak Boukharouba (2017) use support vector machines (SVM) for the recognition of handwritten Persian numerals [2]. Hidden Markov Models (HMM) have also been used in the literature [15]. Machine learning and deep learning techniques show an improved accuracy with respect to the traditional morphological method.

A survey on handwritten digit recognition systems with recent techniques with three well known classifiers namely MLP, SVM and K-NN with feature extraction methods showed that MLP was found to provide the best accuracy. A good accuracy was also observed in diagonal based extraction with MLP and distance profile based extraction with SVM with reduced training time [9]. A comparative analysis of some of the most widely used machine learning algorithms like SVM, KNN and RFC with deep learning algorithms like multilayer CNN has been applied for handwritten digit recognition. Multilayer CNN produced an accuracy of up to 98.70% [3, 8, 19]. MLP was found to be the best classifier in accuracy of recognizing Indian scripts [21].

Ensemble classification techniques have been used in the recent past by researchers for identification of handwritten Persian and Arabian numerals and also handwritten characters. Their approaches were based on PCA, neural networks, KNN and SVM. Their approaches produce an average up to 98% and the computation time is large in terms of minutes. In conclusion, traditional techniques are complex and time consuming. The maximum accuracy that could be achieved using the state-of-the-art methods was 98.7%. Machine learning techniques also consumed longer time. However, in the proposed approach we could achieve an accuracy of 99.17% with less computation time, thus showing a performance increase over the state of the art approaches [10, 11, 13, 22]. A comparative approach of machine learning and deep learning algorithm is proposed in recognizing hand written digits with an accuracy of 97.6% [14]. An approach proposed by Shamim *et al.*, achieved an accuracy of 90.37% by using multilayer perceptron for Austrian handwritten digits [15]. Assegie and Nair (2019) used Decision Tree classification and achieved an accuracy of 83.4% using Kaggle dataset [20].

II. MATERIALS AND METHODS

The proposed study is based on the identification of handwritten digits using classification algorithms. The Pendigits dataset from the UCI machine learning repository has been used for this work. The dataset contains the contribution of 44 writers. 250 samples were collected from 44 writers. The digits were written using a WACOM PL-100V pressure sensitive tablet with an integrated LCD display and a cordless stylus. The writers were asked to write 250 digits in random order inside boxes of 500 by 500 tablet pixel resolution. The (x, y) coordinate information was obtained for the digits 0 to 9. The quality of features or attributes plays an important role in improving the performance of a classifier. The selected dataset is a sixteen dimensional dataset consisting of 10992 instances with 10 classes.

Each attribute has integer value ranging from 0 to 100. Out of the 10992 instances, 67% was used as the training dataset and the remaining 33% was used as the test dataset.

Recognition of handwritten digits or characters is a challenging task since the writing styles and patterns of individual writers are never the same. Moreover, there are similarities between the numerals which makes the recognition process more challenging. Various techniques had been discussed in the literature for the recognition of handwritten characters but the accuracy is poor in many such techniques. Moreover, comparatively less works are available in literature in this field. This paper considered handwritten digit recognition as one of the crucial problems in which machine learning techniques can be applied in order to achieve a good accuracy rate.

A. Classification

Classification is a supervised learning process which identifies the class variable based on a model built using a training set of known class variables. In supervised learning, the prediction is based on a set of finite labels. The machine learns to classify or predict based on a set of known labels. Regression is closely related to classification. In regression, the goal is to predict a continuous target variable whereas in classification the target class has finite labels. The dataset is usually classified into a training set and a test set. A model is created by training the classifier on the training set. Then the model is applied to the test set to predict the class variable. The accuracy of prediction of the classifier on the test set is determined and is considered to be a significant measure of the performance of the classifier.

B. Multivariate Classification

Classification can be of two types namely, binary classification and multiclass classification. A majority of the classification problems fall in the category of binary classification in which the class label consists of only two categories or values. Another type of classification problem is the multiclass classification problem or multivariate classification. In this category the labels consists of multiple values instead of just two labels. In other words, a multiclass classifier classifies data using multiple predetermined labels. Many real life applications are multiclass classification problems. All classification algorithms cannot be applied or suitable for multiclass classification problems. Therefore, a set of suitable algorithms were chosen and analyzed for the proposed work. Moreover, in multiclass classification problems, individual accuracy is not an efficient metric for measuring the performance. Therefore, metrics such as log-likelihood loss or maximum probability are used to measure the performance. The present work uses maximum probability as the metric for classifier performance.

C. Classification Algorithms

The present work used only one individual classification algorithm i.e. RPART. Some individual algorithms were not found to be suitable for the selected dataset since the class labels were numeric.

(i) **RPART**: RPART is a recursive partitioning algorithm. It works by splitting the dataset recursively. The subsets are further split into subsets until a termination criterion

is met. In every iteration, the split is made based on the independent variable and this reduces the heterogeneity of the predicted variable. The split is made using impurity of the predicted node as the metric. Since impurity is a measure of the heterogeneity of the node, it helps in reducing the predicted variable's heterogeneity. Rpart uses entropy and gini index as the impurity quantification methods. The algorithm makes a locally optimized decision at each stage. In some cases, this algorithm settles at a local optimum instead of finding a global optimal tree.

D. Ensemble Algorithms

Ensemble is a classification technique in which multiple classifiers are trained and their predictions are combined as a single classifier. Ensemble techniques are used to improve the performance of weak classifiers especially the multiclass classifiers.

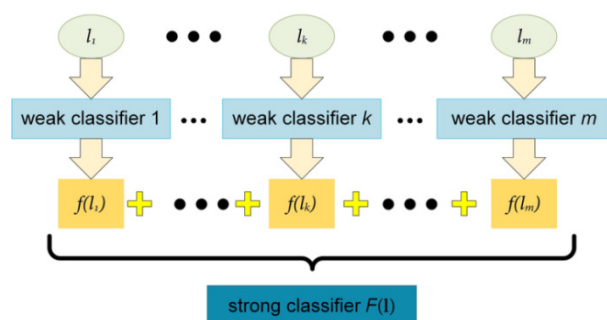


Fig. 1. The Ensemble Process.

The ensemble strategy iteratively learns a set of weak models on subsets of data and weighs each weak prediction according to each weak learner's performance. Then, the predictions of all the weak learners are multiplied by their weights to obtain a final weighted prediction that is expected to be better than the individual predictions. Ensemble can combine two or more algorithms of similar or dissimilar types. Since each of them have a different understanding about the dataset, the final decision would be more accurate, robust and less biased. The present work uses ensemble as one of the techniques for classification in order to improve the performance of the classifiers on recognition of handwritten digits. Fig. 1 shows the ensemble process.

In ensemble classification, the training set is trained with a set of m classifiers $\{c_1, c_2, \dots, c_m\}$ to produce the models $\{f(l_1), f(l_2), \dots, f(l_m)\}$. The test data set is then tested with the new models and fed as input to a meta classifier which makes the final voting using averaging or majority voting. Bagging, boosting and stacking are common ensemble methods.

(i) Random Forest: Random forest is a tree based algorithm. It is a bootstrapping algorithm. Tree based models are known to predict data with higher accuracy and stability. Random forest is a versatile machine learning technique that can be used for classification. Its features include dimensionality reduction and treating of missing values. It is an ensemble method combining weak models to create a powerful model. In this method, multiple trees are grown. Each tree gives a vote for the class. The forest chooses the classification based on majority voting. Each tree is grown to the

largest possible extent and the trees are not pruned. The samples in the training set are chosen with choosing a random of N samples with replacement. Though random forest is good at classification, it is not suitable for regression problems.

(ii) Bagged CART: Classification and Regression Tree (CART) is another decision tree based classification algorithm. CART is suitable both for classification and regression problems. Bagging is another ensemble technique that is used to improve the performance of weak classifiers. Bagging or Bootstrap aggregating is designed to improve the stability, predictability and accuracy of machine learning (22). Bagging is a powerful ensemble method which selects n random samples with replacement. In bagging, a sample of observations are chosen from the original set. Each row is selected with replacement from the original dataset so that each row is less likely to be chosen in the next iteration. This helps to form multiple bootstrapped samples. Then, majority voting or averaging is used to make the final prediction. A bagged CART produces an ensemble of CART using bagging. Bagging allows the trees to grow to full depth without pruning like Random forest. Therefore, these trees produce high variance and low bias at the same time they avoid overfitting.

(iii) Gradient Boosting: Boosting is another ensemble strategy in which new models are combined together to reduce the errors made by existing models. As the name indicates boosting helps to boost the accuracy of weak learners. Boosting assigns variant weights to the classifiers and combines them. Models are added to the existing models until the improvement stops. Boosting is a sequential technique in which the first algorithm is trained on the entire dataset and the other algorithms are built by fitting the residuals of the first algorithm. More weights are given to those observations that are poorly predicted by the previous model. Thus, a series of weak learners are created, each of which might not be good for the entire dataset but are good for a partial dataset. Then, the results are combined. Gradient boosting is an implementation of boosting. Here, new models are created to predict the errors of prior models and then combined together to make the final prediction. It uses the gradient descent algorithm to minimize the loss during the addition of new models. Boosting may lead to over fitting of data if not stopped at the right point.

(iv) Extreme Gradient Boosting: Extreme gradient boosting shortly known as XGBoost, is a gradient boosted decision trees algorithm designed for speed and performance. This algorithm is efficient in computation time and also makes the best use of memory resources. The parallel processing under the hood accounts for the decrease in computation time for this algorithm. It works much faster than other gradient boosting algorithms. It has better scalability too. It is built on gradient boosting framework but more efficient than gradient boosting. The success rate of XGBoost classification is attributed to parallel processing, tree pruning, handling missing values and regularization to avoid overfitting and bias. XGBoost consistently outperforms single performance classification algorithms. XGBoost provides an optimal classification model and prove to be a suitable algorithm when the training samples are large.

E. Experimental Setup

The Pendigits dataset is downloaded from the UCI machine learning repository and it is used for the experiments. This dataset is chosen for the experiments since it contains a large number of instances with simple and valid features. The data was preprocessed to treat missing values. The class label was converted to factor as was required in some of the algorithms. Matrix representation of the datasets were also created. The dataset was divided into two sets, namely, training set and test set. 67% of the dataset was chosen as the training set and the remaining 33% was chosen as the test set.

(i) Model Construction: Initially the algorithms were trained with the training set and the respective classification models were constructed. Ten-fold cross validation was used to improve accuracy and to reduce the classification errors. The model is then tested with the test set.

(ii) Evaluation Metrics: The models were evaluated with accuracy, computation time. Classification accuracy is the ratio of the number of correct predictions made to the total number of predictions.

$$\text{Accuracy} = \text{CP/T} \quad (1)$$

Where CP is the number of correct predictions and T is the total number of predictions. The algorithms were also evaluated with performance metrics such as precision, recall, F-score, Kappa coefficient and computation time.

(iii) Comparison of Results: The models were constructed using different algorithms and the test set was tested with the different models. The performance of the algorithms was compared and XGBoost was found to outperform the various algorithms.

(iv) Parameter Tuning: The performance of the XGBoost algorithm was further improved by tuning the hyper parameters. Parameter tuning was used to find the optimum values for the parameters for identifying the handwritten digits with higher accuracy. The performance of the XGBoost classifier was improved by tuning of the following parameters:

— Learning Rate: The learning rate is the step size with which the gradient is descended. It makes the model robust by shrinking the weights at each step. The default learning rate value is 0.3.

— Maximum Depth: Maximum depth refers to the maximum depth of a tree. This parameter is used to control overfitting. The default value is 6.

— Minimum Child Weight: Minimum child weight refers to the sum of weights of all observations required in a child. This parameter is also used to control overfitting. The default value is 1.

— Subsample: Subsample is the fraction of observation to be used as samples for each tree. Lower values of subsamples can prevent overfitting. Too small values may lead to under fitting. The default value is 1.

— Column Sample: This is the fraction of columns to be used as the sample. The default value is 1.

III. RESULTS AND DISCUSSION

A. Performance Evaluation

The classification accuracy of each algorithm was evaluated as the first performance metric for the classifiers. The confusion matrix was constructed and the results were obtained. Confusion matrix is a clear

and unambiguous way of presenting the classification results. A sample confusion matrix that is obtained for the XGBoost algorithm is shown in Fig. 2.

372	0	0	0	0	0	0	0	2	0
0	396	4	0	1	0	0	0	0	2
0	2	364	0	0	0	0	0	0	0
0	0	0	349	0	0	0	1	0	0
1	0	0	0	414	0	0	0	0	0
0	0	0	1	0	352	0	0	1	1
0	0	1	0	0	0	314	0	0	0
0	1	2	1	0	0	0	392	1	0
1	1	0	0	0	1	0	1	316	0
0	0	0	1	0	1	0	0	2	329

Fig. 2 Confusion Matrix for the XGBoost classifier.

Fig. 3 compares the classification accuracy of various classifiers. The accuracy was measured in percentage and plotted.

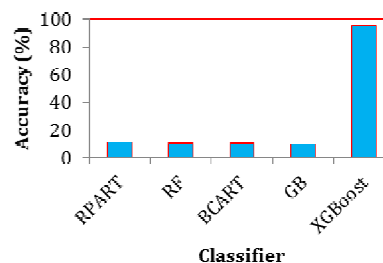


Fig. 3. Accuracy of the Classifiers.

The results show that the extreme gradient boosting algorithm outperforms the other classifiers with a very high accuracy of 95.67%. Precision and recall are two more metrics that are used to measure the performance of the classifiers. These are metrics which help to measure the relevance of a classifier. Precision which is also known as positive predictive value is the fraction of relevant instances that have retrieved to the total number of relevant instances. Precision is a measure of the exactness of a classifier.

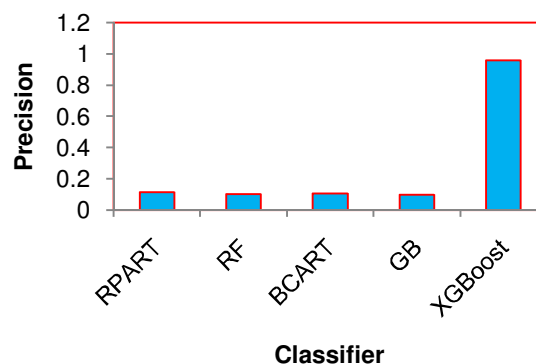


Fig. 4. Precision values of the algorithms

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \quad (2)$$

Where TP represents True Positives and FP represents False Positives. A higher precision indicates higher relevance. Fig. 4 plots the precision of each of the algorithms. These results indicate that the extreme gradient boosting is relevant also for the chosen dataset than the other classifiers.

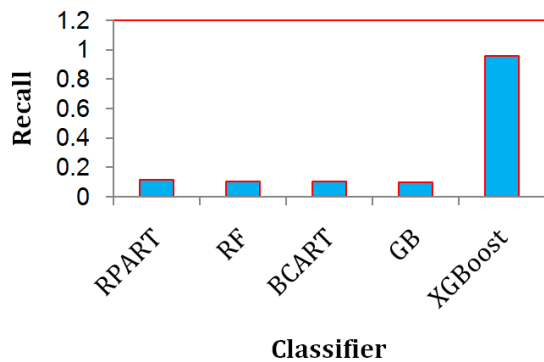


Fig. 5. Recall value of the Classifiers.

Recall is the ratio of the number of true positives to the sum of true positives and false negatives. Recall is a measure of a classifier's completeness.

$$Recall = TP / (TP + FN) \quad (3)$$

A lower value of false negatives or irrelevant data can also produce a higher recall. Precision and recall together can be used to evaluate the relevance of a classifier. Fig. 5 shows the recall of the classifiers considered for this study. F-score or F-measure is another metric used for evaluating the performance of the classifiers. It specifies the balance between the precision and recall. The formula for F-score is given in Eqn. (4).

$$F\text{-Score} = (2 * ((precision * recall) / ((precision + recall))) \quad (4)$$

Fig. 6 shows the F-measure of the classifiers. The F-measure also indicates that the extreme gradient boosting is the best classifier for a multiclass dataset. The computation time for the different classifiers were also evaluated since it is one of the major criteria for the efficiency of classifiers. The computation time was calculated as an average of 100 runs. The computation time of various algorithms is calculated and plotted in Fig. 7.

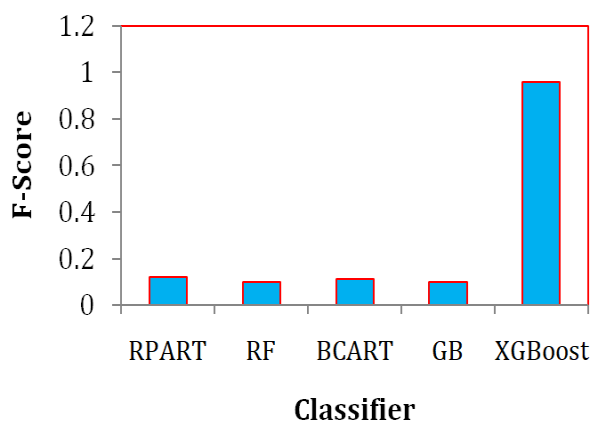


Fig. 6. F-Score of the Classifiers.

The computation time for construction of the model was calculated in seconds. The results show that the random forest and gradient boosting took more time for model construction. The XGBoost algorithm seems to be far more efficient than the other classifiers. RPART consumed the least computation time. This may be because it is an individual algorithm. Out of the

ensemble algorithms, Bagged CART was the best in terms of computation time followed by XGBoost.

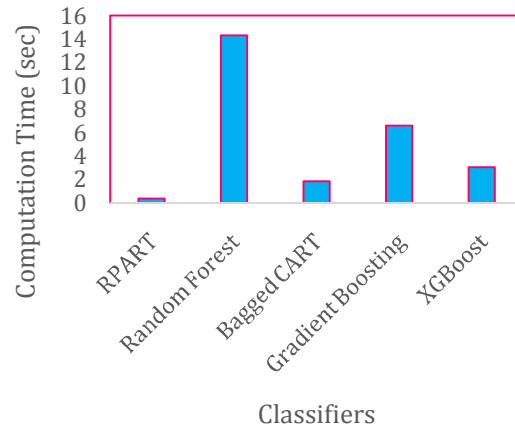


Fig. 7. Computation Time.

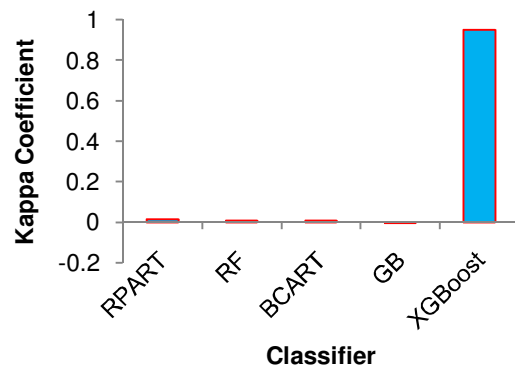


Fig. 8. Kappa Coefficient.

Kappa statistics is a measure of interrater reliability. It measures the extent to which the data collected in the study are correct representations of the measured variables. Kappa coefficient is acceptable if its value is above 0.4. The results are shown in Fig. 8. The results show that only for the extreme gradient boosting the kappa coefficient is in the acceptable range.

B. Improving Performance using Parameter Tuning

The performance of the XGBoost classifier was further improved by tuning the hyper parameters. The learning rate of the classifier has a default value of 0.3. In order to tune the learning rate, higher learning rates were applied to the classifier starting from 0.5 and then the learning rate was reduced in steps of 0.05. The accuracy got improved to 99.17% when the learning rate is 0.38.

Fig. 9 shows the improvement of accuracy through tuning of the learning rate. The next parameter that has been tuned was maximum depth.

The maximum depth was changed from 1 to 6 in steps of 1 at three different learning rates for which the accuracy was maximum. In all three cases, the optimum value of the maximum depth was found to be 3.

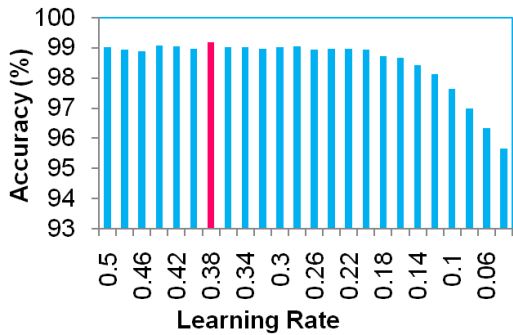


Fig. 9. Tuning of Learning Rate

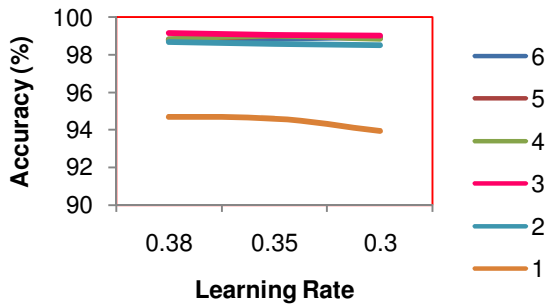


Fig. 10. Tuning of maximum depth.

Fig. 10 shows the tuning of the maximum depth. It has been found that the accuracy reached it highest value of 99.17 when the learning rate was 0.38 and the maximum depth was 3.

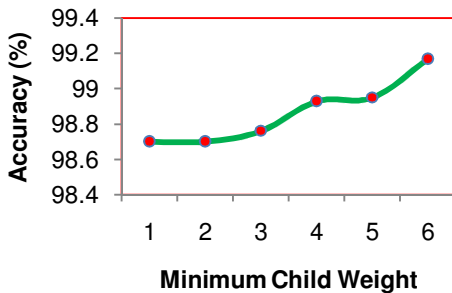


Fig. 11. Tuning of Minimum Child Weight.

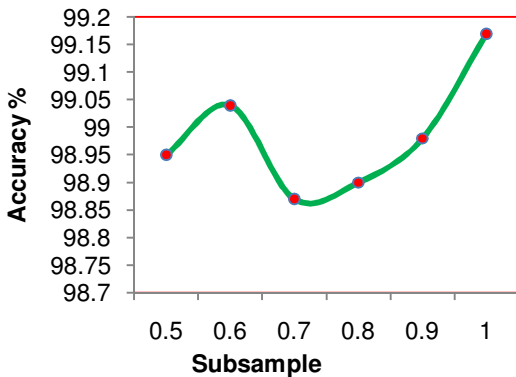


Fig. 12. Tuning of Subsample.

Fig. 11 shows the tuning of minimum child weight. It was tuned with values from 1 to 6 in steps of 1 and the optimum value was found to be 1 at learning rate 0.38. Subsample refers to the frequency of observations to be used as samples for each tree. The subsamples were tuned with values from 0.5 to 1 in steps of 0.1 and it was found that the highest accuracy was obtained for value 1. Fig. 12 shows the tuning of the subsample.

The last parameter to be tuned was column sample. This parameter was also tuned with values from 0.1 to 1 in steps of 0.1. The optimal value was found to be 1. Fig. 13 shows the tuning of column sample.

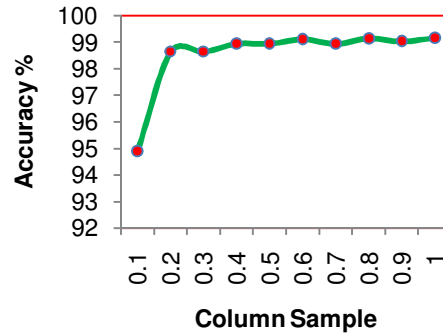


Fig. 13. Tuning of column sample.

It has been inferred from the experiments that parameter tuning can improve the performance of the XGBoost classifier to a significant extent. The accuracy was improved from 95.67% to 99.17%. The optimal values were found as 0.38 for learning rate, 3 for maximum depth and 1 for minimum child weight, subsample and column sample.

IV. CONCLUSION

This work presents a machine learning approach for improving the classification accuracy of recognition of handwritten characters using ensemble techniques. Indian (Arabic) numerals were used for this research. The performance of five classification algorithms were evaluated using different evaluation metrics. It has been inferred that the extreme gradient boosting algorithm outperforms all other classifiers in terms of almost all performance metrics. It has been concluded that the extreme gradient boost algorithm is suitable for handwritten digit recognition and also for multiclass classification problems. The accuracy of the XGBoost algorithm was further increased by tuning the hyper parameters. An accuracy of 99.17% could be achieved using the proposed approach and this is considered to be the best accuracy for this problem. Moreover, the proposed approach could achieve better accuracy in less computation time. The computation time is reduced by up to 3.07 seconds.

V. FUTURE SCOPE

Future works include testing the performance of XGBoost algorithm by optimizing the parameters for other multiclass classification problems and also evaluating the performance of the classifier for handwritten character recognition. We are also working

with deep learning techniques for identifying handwritten digits and characters.

ACKNOWLEDGMENTS

The authors acknowledge the UCI machine learning repository for providing the necessary permissions for using the dataset.

Conflict of Interest. There are no conflicts of interest.

REFERENCES

- [1]. Boukharouba, A., & Bennia, A. (2011). Recognition of handwritten Arabic literal amounts using a hybrid approach. *Cognitive Computation*, 3(2), 382-393.
- [2]. Abdelhak Boukharouba, & Abdelhak Bennia, (2017). Novel Feature Extraction Technique for the Recognition of Handwritten Digits. *Elsevier Applied Computing and Informatics*, 13(1): 19-26.
- [3]. Anuj Dutt, AashiDutt, (2017). Handwritten Digit Recognition using Deep Learning. *International Journal of Advanced Research in Computer Engineering & Technology*, 6(7): 990-997.
- [4]. Badr B. Al, & Haralick R. M. (1998). A Segmentation free approach to text recognition with application to Arabic Text. *Proceedings of the Third International Conference on Document Analysis and Recognition, IEEE Computer Society*, 1(3): 147-166.
- [5]. Álvarez, D. C., Rodríguez, F. M., & Hermida, X. F. (1998). Printed and handwritten digits recognition using neural networks. In *Proceedings of ICSPAT-98 (International Conference on Signal Processing and Applications Technology)*, 1-5.
- [6]. F.A. Al-Omari, & O. Al-Jarrah, (2004). Handwritten Indian numerals recognition system using probabilistic neural networks. *Advanced Engineering Informatics*, 18(1): 9–16.
- [7]. Alwzwy, H. A., Albehadili, H. M., Alwan, Y. S., & Islam, N. E. (2016). Handwritten digit recognition using convolutional neural networks. *International Journal of Innovative Research in Computer and Communication Engineering*, 4(2), 1101-1106.
- [8]. Haralick Robert M., & Kanungo Tapas, (1990). Character recognition using mathematical morphology. *Proceedings of the United States Postal Service Advanced Technology Conference*, 2(1): 973–986.
- [9]. Ishani Patel, Virag Jagtap, & Ompriya Kale, (2014). A Survey on Feature Extraction Methods for Handwritten Digits Recognition. *International Journal of Computer Applications*, 107(12): 11-17.
- [10]. Kumar, J., & Chakravarthy, V. S. (2008). Designing an optimal Classifier Ensemble for online character recognition using Genetic Algorithms. *11th International Conference on Frontiers in Handwriting Recognition*, 9-14.
- [11]. Kathirvalavakumar Thangairulappan, & Palaniappan Rathinasamy (2016). Ensemble Neural Network in Classifying Handwritten Arabic Numerals. *Journal of Intelligent Learning Systems and Applications*, 8(1): 1-8.
- [12]. Kim, J. W., Kim, K. I., Choi, B. J., & Kim, H. J. (1999). Decomposition of Chinese character into strokes using mathematical morphology. *Pattern Recognition Letters*, 20(3), 285-292.
- [13]. Kazemi, M., Yousefnezhad, M., & Nourian, S. (2015). A new approach in Persian handwritten letters recognition using error correcting output coding. *Journal of Advances in Computer Research*, 6(4), 107-124.
- [14]. Meer Zohra, D. & Rajeswara Rao, (2019). A Comprehensive Data Analysis on Handwritten Digit Recognition using Machine Learning Approach. *International Journal of Innovative and Exploring Engineering*, 8(6): 1449-1453.
- [15]. S. M. Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana & Abdullah Al Jobair, (2018). Handwritten Digit Recognition using Machine Learning Algorithms. *Global Journal of Computer Science and Technology Deep Neural & Artificial Intelligence*, 18(1): 1-9.
- [16]. S.M. Awaidah, S.A. Mahmoud, (2009). A multiple feature/resolution scheme to Arabic (Indian) numerals recognition using hidden Markov models. *Elsevier Signal Processing*, 89(6): 1176–1184.
- [17]. Saeed AL-Mansoori (2015). Intelligent Handwritten Digit Recognition using Artificial Neural Network. *International Journal of Engineering Research and Applications*, 5(5): 46-51.
- [18]. SkMd Obaidullaha, ChayanHalderb, NibaranDasc and Kaushik Royb, (2015). Numeral Script Identification from Handwritten Document Images. Eleventh International Multi-Conference on Information Processing, *Elsevier Procedia Computer Science*, 54: 585-594.
- [19]. T. Siva Ajay (2017). Handwritten Digit Recognition Using Convolutional Neural Networks. *International Research Journal of Engineering and Technology*, Vol. 4(7): 2971-2975.
- [20]. Assegie, T. A., & Nair, P. S. (2019). Handwritten digits recognition with decision tree classification: a machine learning approach. *International Journal of Electrical and Computer Engineering*, 9(5), 4446-4451.
- [21]. Kumar, V. V., Srikrishna, A., Babu, B. R., & Mani, M. R. (2010). Classification and recognition of handwritten digits by using mathematical morphology. *Sadhana*, 35(4), 419-426.
- [22]. Anugrahanti, W. W., Wibowo, A., & Meilanani, S. (2017). Bagging Approach for Increasing Classification Accuracy of CART on Family Participation Prediction in Implementation of Elderly Family Development Program. *Health Notions*, 1(2), 99-104.
- [23]. Perwej, Y., & Chaturvedi, A. (2011). Machine Recognition of Hand Written Characters using Neural Networks. *International Journal of Computer Applications*, 14, 6-9.

How to cite this article: Beulah, C. B. C., Sujni P., and Carolin, S. (2020). Improving the Accuracy of Handwritten Digits Recognition through Parameter Tuning of XG Boost Algorithm. *International Journal on Emerging Technologies*, 11(4): 116–122.