

Analysis of machine learning algorithms for character recognition: a case study on handwritten digit recognition

Owais Mujtaba Khanday, Dr. Samad Dadvandipour

Institute of Information Sciences, University of Miskolc, Miskolc, Hungary

Article Info

Article history:

Received Jun 5, 2020

Revised Aug 3, 2020

Accepted Aug 16, 2020

Keywords:

Handwritten digit recognition

MLP

OCR

Random forests

SVM

ABSTRACT

This paper covers the work done in handwritten digit recognition and the various classifiers that have been developed. Methods like MLP, SVM, Bayesian networks, and Random forests were discussed with their accuracy and are empirically evaluated. Boosted LetNet 4, an ensemble of various classifiers, has shown maximum efficiency among these methods.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Owais Mujtaba Khanday

Institute of Information Sciences, The University of Miskolc

Egytemvaros, 3525, Miskolc, Hungary

Email: aitowais@uni-miskolc.hu

1. INTRODUCTION

Significant achievements have been made in the optical character recognition (OCT) technology, including handwritten recognition of digits. Its role is ubiquitous in real-time e-processing of the data such as reading the zip codes and sorting the post mails, bank check processing, e-commerce and even student achievement recognition, etc. [1, 2]. Considerable progress has been achieved because of the developments and advancements in the computational power of computers and the availability of more massive datasets that are used for training and testing purposes [3]. Nowadays, handwritten recognitions are even used for communication purposes. OCT does have the capability and power to pillar the paperless environment by processing the existing paper documents [4]. Handwritten digit recognition involves identifying 10 characters, i.e., 0-9, but the input is sensitive to the environmental noise. [5, 6]. One of the foremost tasks is to identify the local areas for obtaining discriminating features. Various sampling techniques have been developed to finding these local regions [7]. Also, the datasets are even vague because the writing and orientation differ from person to person. It also happens sometimes; one cannot even recognize the handwritten characters written by himself. Other problems include slipping of the pen, letter insertion, or omission, which greatly complicate the task [8]. However, a typical recognition system is built to focus on only a subset of the problem [9].

The most common method for building the handwritten digit recognizer is using a multi-layer perceptron (MLP) [10, 11] kNN and support vector machine SVM. Various other techniques were developed using different techniques with MLP structure. Renata F.P Neves and et al. proposed a method for handwritten digit recognition, which improved the efficiency rates compared to the MLP and hybrid

classifiers over the NIST SD19 digit database [12]. Cheng-Lin Liu and et al. combined eight classifiers and with ten feature vectors. They CENPARMI, CEDAR, and MNIST databases were tested. SVC with RBF kernel (SVC-RBF) gave the highest accuracy in most cases but on the cost of storage and computation [13]. A study was done in [14] showed the bundle of feature extraction techniques and were evaluated using the benchmark datasets available publically; the methods [15] and [16] outperformed the other methods available in the literature by showing the accuracy of 99.03% and 98.75% respectively. The following research methods are applied in the paper.

2. RESEARCH METHOD

2.1. Multi-layer perceptrons

An MLP is an artificial neural network classifier that employs the modeling of human biological neurons [17]. It is a feedforward network that computes the sigmoid function of the weighted sum of all the input neurons, as given in (1). The typical MLP looks like as given in Figure 1.

$$\frac{1}{1+e^{-\sum_k w_k x_k + b}} \tag{1}$$

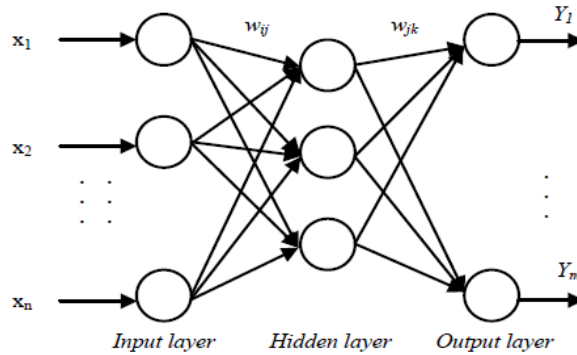


Figure 1. Architecture of MLP

Where w_i the weight of the input is, x_i is the input coming from the i th neuron, and b is the bias. The sigmoid function is also called as the activation function. The classifier consists of three layers, the input layer, the output layer, and the hidden layer. The network is fully connected, and each layer has a certain number of neurons. The number of input and output neurons depends upon the attributes and the number of classes existing in the data set. In the case of the digit classifier, the output neurons are 10. For any M-class classification, the model has m outputs. On input pattern of $x = (x_1, x_2, x_3, \dots, x_d)^T$ The output of class k is computed by:

$$y_k(x) = S \left[\sum_{j=1}^{N_h} w_{kj} s(V_j^T x + v_{j0}) + w_{k0} \right] = S \left[\sum_{j=1}^{N_h} w_{kj} h_j + w_{k0} \right] \tag{2}$$

N_h is the number of hidden units W_{kj} and V_{ji} are the connecting weights of the output layer and the hidden layer. The MLP model is trained against a test set, and it automatically learns to adjust the weights for each connection [18]. The learning process used is the error back-propagation algorithm, which adjusts the connecting weights by minimizing the mean square error (MSE) over a set of N_x training examples [19].

$$E = \left\{ \sum_{n=1}^{N_x} \sum_{k=1}^M [y_k(X^n, W) - t_k^n]^2 + \lambda \sum_{w \in W} w^2 \right\} \tag{3}$$

- Input X: A set of activations a^1 for the input layer,
- Feedforward: For each $l=2,3,\dots,L$ compute $Z^l = w^l a^l + b^l$ and $a^l = \sigma(Z^l)$
- Output error δ^l : Compute the vector $\delta^l = \nabla_a C \odot \sigma'(Z^l)$
- Back propagate the error: For each $l = L - 1, L - 2, \dots$, compute $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(Z^l)$
- Output: The gradient of the cost function: $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$, and $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

2.2. Support vector machine (SVM)

SVM is a supervised learning model introduced by Joachim’s and subsequently used by others. It attempts to find the dimensional space among all the $|T|$ -dimensional spaces that separate the negative form of the positive training examples [20, 21]. It is based on Structural Risk Minimization [22-24]. High dimensional feature spaces, few irrelevant features (dense concept vector), and sparse instance vectors are the particular properties of the text acknowledged by the SVM [24]. SVMs map data to a high dimensional feature so that the data points could be categorized even the data is not linearly separable. Mathematical functions used by SVMs in IBM® SPSS® for transformation are linear, polynomial, Radial Basis Function, and sigmoid. SVMs undertake two important advantages for TC Joachims [21].

- a) Term selection is often not required, as SVMs tend to be reasonably robust to overfitting and can scale up to considerable dimensionalities;
- b) No human and machine effort in parameter tuning on a validation set is needed because there is a theoretically motivated, “default” choice of parameter settings, which has also been architecture.

There are four types of multi-class architectures using binary classifiers: one-against-rest, one-against-one, acyclic direct graph-ADG, and unbalanced decision tree-UDT [25]. In one against rest architecture for distinguishing, m classes m classifiers are needed. Every classifier C_i is to be trained for recognizing class i . The one-against-one $m(m - 1)/2$ classifiers are needed for each different class pair and are evaluated in parallel. Samples of only i and j are used to train the classifier C_{ij} . When the classifier recognizes the sample x belonging to class i or j a vote is assigned to i or j respectively. After all the classifiers have classified the sample x the class which received the more votes is considered to be the class to which sample x belongs. Figure 2 shows all the four combinations in which SVMs can be architecture for the classifier. The number of classifiers needed for the four different types of multi-class architectures is listed in Table 1.

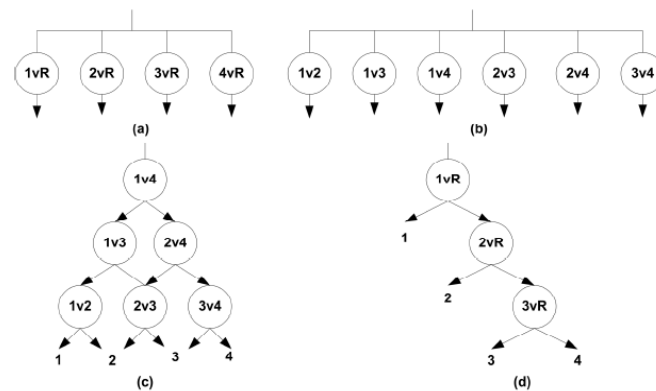


Figure 2. (a) one against rest, (b) one against one, (c) cyclic direct graph, (d) unbalanced decision tree

Table 1. Binary classifier architectures

Architecture	Number of Classifiers	Classifiers used for Sample Classification
one against one	$m * \frac{m - 1}{2}$	$m * \frac{m - 1}{2}$
one against rest	m	m
ADH	$m * \frac{m - 1}{2}$	$m - 1$
UDH	$m - 1$	$\frac{m - 1}{2}$

2.3. Bayesian networks

A Bayesian network $B = \langle N, A, \theta \rangle$ is a directed acyclic graph (DAG) with a conditional probability distribution (C.P. table) for each node, collectively represented by Θ . Each node $n \in N$ represents a domain variable, and each arc $a \in A$ between nodes represents a probabilistic dependency [26]. This recognition model learns the conditional probability of each attribute A_i from the class label C from the training data. Its performance has been proved to be competitive with state-of-the-art classifiers [27]. The advantages of these models are they can fit complex problems in any domain, whether continuous, discrete,

or mixed, with undetermined labels, partial labels, or many class variables to be simultaneously predicted [28]. The Bayesian classifier represented by a Bayesian network is defined as follows:

$$c(E) = \arg \max_{c \in C} P(c)P(a_1, a_2, \dots, a_n | c) \quad (4)$$

Assuming that all the attributes are independent given the class. The conditional independence assumption is [30]:

$$P(E|c) = P(a_1, a_2, \dots, a_n | c) = \prod_{i=1}^n P(a_i | c) \quad (5)$$

The resulting naïve Bayesian classifier is:

$$c(E) = \arg \max_{c \in C} P(c) \prod_{i=1}^n P(a_i | c) \quad (6)$$

The main problem in confronting naïve Bayes is its conditional attribute assumption; Naïve Bayes classification improves the accuracy in the domains with irrelevant or redundant attributes but not in the others. Thus, a variety of methods were developed for improving its efficiency; for example, the tree augmented naïve Bayes [29] leads to acceptable computational complexity. The other approaches used by Liangxia and et al. for improving its efficiency are feature selection, structure extension, local learning, and data expansion. The improved algorithms used are ENB, SP-TAN, LWNB (K=50), and LNB. ENB and SP-TAN outperformed N.B. increasing the efficiency from 82.41% to 83.22 and 84.76%, respectively, in the 36 data sets that were tested. [30].

2.4. Random forests

Random Forest is a term for classifier combinations, also known as Classifier Ensembles, i.e., a combination of Multiple Classifier Systems (MCS) to improve the reliability in comparison with individual classifiers [31]. The classifier uses L tree-structured classifiers $\{h(x, \Theta_k), k = 1, \dots, L\}$, where x is the input, and Θ_k are independently identically distributed random vectors. Thus, it is said a family of methods having various algorithms. The four approaches proposed for building the MCPs are the design level, the classifier level, the feature level, and the data level [32]. The last two approaches have been used and proven exceptionally successful involving the Bagging technique, boosting technique, and Random subspace principles [32-34]. In the Random Subspace principle, each tree is grown as below:

- a) If there are N training examples in a training set, then sample N cases for the resulting training set of the tree at random without replacement.
- b) A number $K < M$ where M is the number of features is set at each node. A subset of the K is chosen at random, and among them, the best split is selected.
- c) Each tree is grown to its maximum size and unpruned.

The algorithm works on two parameters L and K; L: number of trees in the forest and K features that are preselected for the splitting process. Many researchers have used R.F. Bermin [35] split at each node is done according to the linear combinations of the features instead of a single one. Robnik [36] improved the combination process by introducing the weighted voting method. Boinee et al. introduced Meta Random Forests; it consisted of R.F. as base classifiers for combination techniques. In almost many algorithms, the number L; no of trees in the forest has been arbitrary chosen equal to 100. Bermain chose $K = \log_2 M + 1$, without explaining the reason.

Simon Bernard et al. [31] experimented on MNIST dataset and tried to explain the parametrization influence for the R.F. The number K of features randomly selected in the Forest-RI process has to be greater than 1, but its value can not be too high as the accuracy tends to converge with increasing but stops as $K > 12$. All curves for values $K = 1 - 6$ begin to rise and are constant till the value of $K = 20$ and then start to decrease and reach a minimum at $K = 84$, which could be because too many features preselected makes the diversity decrease between trees in forests. The more the features are randomly selected, the more the trees are identical to each other. Recognition rates with respect to K and L are shown in Figure 3, and we can conclude that in the tree inducing process, rather than randomly choosing the splitting tests, then we should implement some selection measure. The recognition rate maxima in the experiment conducted by the Simon [31] in case of handwritten digit recognition, the maxima recognition rates are reached in the area defined by the intervals [100, 300] for L and [5, 20] for K.

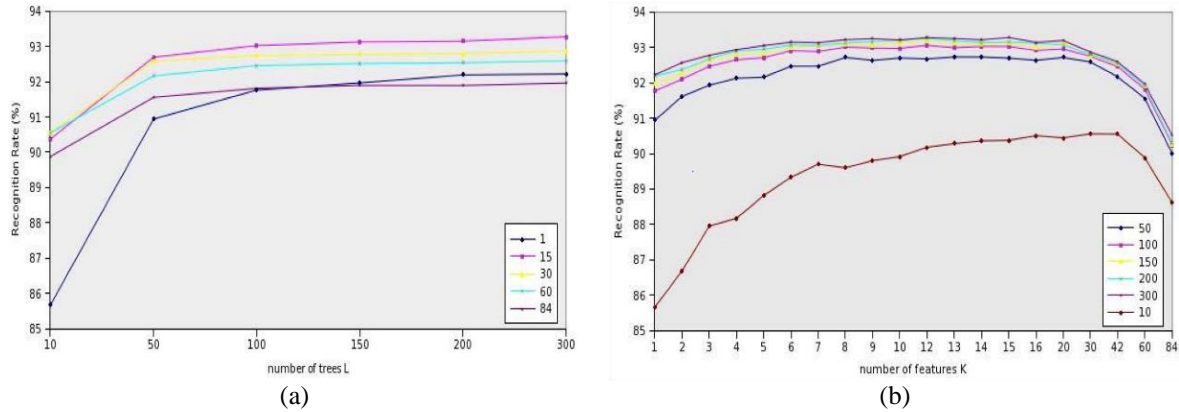


Figure 3. Recognition rates with respect to L and K [31] a) Recognition Rates wrt L, b) Recognition Rates wrt K

3. RESULTS AND DISCUSSION

MLPs are efficient classifiers for the multi-class problem, but using a back-propagation algorithm as the learning algorithm is a disadvantage because the algorithm could stop at a local minimum. However, a momentum strategy could be used for escaping the local minimum, but then it overfits the weights, thus decreasing the generalizing capability [37]. Figure 4 shows the various types of MLPs used by the various researchers [3] for digit recognition. The y-axis is the error percentage, and the x-axis represents the different types of classifiers used. Comparison of different classifiers, as shown in Table 2.

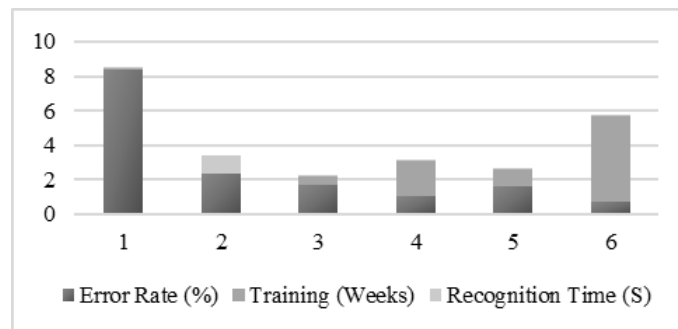


Figure 4. Classifiers with their training time, error, and recognition rates

It is evident that to access a meager error rate, and there is a need for lots amount of time to be spent by the model for training, i.e., lots of data was provided for the training purpose. It is seen with increased accuracy, and there is a proportional increase in training time. It follows the simple rule with the more experience that you learn more [38].

Table 2. Comparison of different classifiers

Method	Error Rate (%)	Training (Weeks)	Recognition Time (S)
Baseline linear Classifier	8.4	0.07	0.01
Baseline nearest neighbor (k=3)	2.4	0	1
LetNet1	1.7	0.5	0.015
LetNet4	1.1	2	0.03
Fully connected multi-layer neural networks	1.6	1	0.1
Boosted LetNet4	0.7	5	0.05

Best margins of separation between any two classes are found by the SVM thus, considered as the best binary classifier. It finds the best or optimal liner function that separates the classes. Renata F.P et al.

used an SVM based recognizer that uses 45 SVMs for classification because the SVM is a binary classifier, and there are ten output classes for this problem, so the architecture must be composed of 45 SVMs to compare for all the combinations. The hyperbolic tangent function is used as an activation function and gradient descent algorithm for learning. The method showed better performance on the NSID SD19 dataset than MLP, 10 SVMs, and Bellili-MLP+15/45 SVM. The error rate is reduced to 2.06 %. Comparison of error rates of different classifiers, as shown in Figure 5.

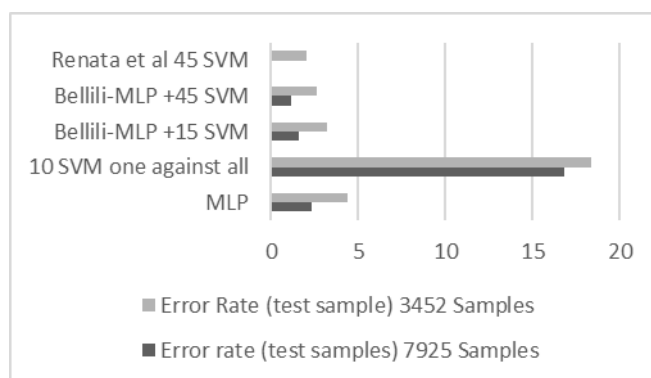


Figure 5. Comparison of error rates of different classifiers

Kandming Liu proposed a model on a Bayesian network with the concavity feature and classification and association rules for digits to select more features that effectively achieve a more accurate result. The training sample consisted of 600 single digit samples selecting 20 non-overlapping digit samples for test and attained the accuracy of 65.07 total percent and 34.93% of false rate on 200 samples. Breiman proposed a model in [35] called Forest-RC, where at each node, a split is done based on the linear combination of features, which allows the dealing with the cases having only a few inputs supplied. For similarity evaluation, the accuracies on similar inputs were examined, so to remove from the vote that shows the lowest values.

Robert in [36] improved the combination process due to original Forest-RI by introducing the weighted voting method to take into account a restricted subset of the classifier outputs based on the individual accuracies of the similar instances. Another method for R.F. is called Meta Random. Boniee et al. proposed Forests [39] in which R.F. as a base classifier of combination techniques were used. None of the methods can be proven to be superior to others in terms of accuracy, plus they have not given experimental work from a practical point of view. It could be seen in these methods some of the parameter values were commonly used like L. The number of trees in forests was fixed to 100. Bernard et al. [40] tried to understand the roles of the parameter values on the behavior of R.F. Concerning the increase in the number of trees, the accuracy of R.F. converges, and the increase in the recognition rate slows down after 100 trees, and the increment was logarithmic. They confirmed that the tree inducing process should implement a splitting selection measure rather than randomly choosing splitting tests. However, it can be seen that the recognition rate does not increase beyond 93.51%.

4. CONCLUSION

Handwritten digit recognition is a significant field, much progress has been made, and many techniques have been developed. The boosted LeNet 4 method performs the best with the accuracy of 99.3% and is the best among the methods that have been studied in this paper. The only tradeoff is the training time, which is very large and is about five weeks. The operational/actual recognition time is 0.05 ms.

ACKNOWLEDGEMENTS

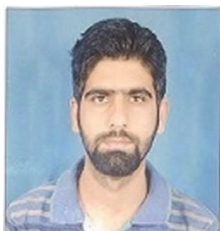
The task was performed under the framework of FIKP higher education institutional excellence program on the optimization of natural resources based on modern technologies within the theme “applying intelligent information tools to involve, reduce and reduce professional losses”.

REFERENCES

- [1] Pedrycz, Witold, Andrzej Skowron, and Vladik Kreinovich, "A Fuzzy Regression Approach to Acquisition of Linguistic Rules," *Handbook of Granular Computing*. John Wiley & Sons, pp. 719-732, 2008.
- [2] Shuying, Yang. "Image Recognition and Project Practice," Beijing: Publishing House of Electronics Industry. 2014.
- [3] Bottou, Léon, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Larry D Jackel, Yann LeCun, et al., "Comparison of Classifier Methods: A Case Study in Handwritten Digit Recognition," *Paper presented at the International conference on pattern recognition*, 1994.
- [4] Shamim, SM, Mohammad Badrul Alam Miah, Masud Rana Angona Sarker, Abdullah Al Jobair, "Handwritten Digit Recognition Using Machine Learning Algorithms," *Global Journal of Computer Science and Technology*, vol. 19, no. 1, pp. 16-23, 2018.
- [5] Basu, S., Das, N., Sarkar, R., Kundu, M., Nasipuri, M., & Basu, D. K., "Recognition of numeric postal codes from multi-script postal address blocks," In *International Conference on Pattern Recognition and Machine Intelligence*, Springer, pp. 381-386, 2009.
- [6] Kumar, Vikas, "Online Handwriting Recognition Problem: Issues and Techniques," *MIT Int. J. Comput. Sci. Inform. Technol.*, vol. 4, no. 1, pp. 16-24, 2014.
- [7] Das, Nibaran, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri, and Dipak Kumar Basu, "A Genetic Algorithm Based Region Sampling for Selection of Local Features in Handwritten Digit Recognition Application," *Applied Soft Computing*, vol. 12, no. 5, pp. 1592-606, 2012.
- [8] Plamondon, Réjean, and Sargur N. Srihari, "Online and Off-Line Handwriting Recognition: A Comprehensive Survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 63-84, 2000.
- [9] Connell, Scott D, "Online Handwriting Recognition Using Multiple Pattern Class Models," *citeseer*, 2000.
- [10] Bishop, Christopher M., "Neural Networks for Pattern Recognition," Oxford university press, 1995.
- [11] Haykin, Simon, "Self-Organizing Maps," *Neural networks-A comprehensive foundation*, 2nd edition, Prentice-Hall, 1999.
- [12] Neves, Renata F. P., Alberto N. G. Lopes Filho, Carlos A. B. Mello, and Cleber Zanchettin, "An SVM Based Off-Line Handwritten Digit Recognizer," *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 510-515, 2011.
- [13] Liu, Cheng-Lin, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa, "Handwritten Digit Recognition: Benchmarking of State-of-the-Art Techniques," *Pattern Recognition*, vol. 36, no. 10, pp. 2271-2285, 2003.
- [14] Soora, Narasimha Reddy, and Parag S. Deshpande, "Review of Feature Extraction Techniques for Character Recognition," *IETE Journal of Research*, vol.64, no. 2, pp. 280-295, 2018.
- [15] Soora, Narasimha Reddy, and Parag S. Deshpande, "Novel Geometrical Shape Feature Extraction Techniques for Multilingual Character Recognition," *IETE Technical Review*, vol. 34, no. 6, pp. 612-621, 2017.
- [16] Soora, Narasimha Reddy, and Parag S Deshpande, "Robust Feature-Extraction Technique for License Plate Characters Recognition," *IETE Journal of Research*, vol. 61, no. 1, pp. 72-79, 2015.
- [17] Das, Nibaran, Ayatullah Faruk Mollah, Sudip Saha, and Syed Sahidul Haque, "Handwritten Arabic Numeral Recognition Using a Multi Layer Perceptron," *Proc. National Conference on Recent Trends in Information Systems*, pp. 200-203, 2006.
- [18] Kruse, Rudolf, Christian Borgelt, Christian Braune, Sanaz Mostaghim, and Matthias Steinbrecher, "Computational Intelligence: A Methodological Introduction," *Springer*, pp. 47-81, 2016.
- [19] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representations by back-propagation errors," *Nature*, vol. 323, no.9, pp. 533-536, 1986.
- [20] Robbins, Herbert, and Sutton Monro, "A Stochastic Approximation Method," *The annals of mathematical statistics*, vol. 22, no. 3, pp. 400-407, 1951.
- [21] Sebastiani, Fabrizio. Consiglio Nazionale Delle Ricerche, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1-47, 2002.
- [22] Vapnik, Vladimir N., "Constructing Learning Algorithms," in *the Nature of Statistical Learning Theory*, Springer, pp. 119-66, 1995.
- [23] Cortes, Corinna, Vladimir Vapnik, "Support Vector Networks "Machine learning," vol. 20, no. 3, pp. 273-97, 1995.
- [24] Joachims, Thorsten, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Paper presented at the European conference on machine learning*, 1998.
- [25] Hassan, Ali, and Robert I. Damper, "Classification of Emotional Speech Using 3dec Hierarchical Classifier," *Speech Communication*, vol. 54, no. 7, pp. 903-16, 2012.
- [26] Cheng, Jie, and Russell Greiner, "Comparing Bayesian Network Classifiers," *Paper presented at the Proceedings of the Fifteenth Conference on Uncertainty in artificial intelligence*, pp. 101-108, 1999, in <https://arxiv.org/ftp/arxiv/papers/1301/1301.6684.pdf>.
- [27] Seewald, Alexander K., "On the Brittleness of Handwritten Digit Recognition Models," *ISRN Machine Vision*, vol. 2012, 2011.
- [28] Bielza, Concha, and Pedro Larrañaga, "Discrete Bayesian Network Classifiers: A Survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1. 2014.
- [29] Friedman, Nir, Dan Geiger, and Moises Goldszmidt, "Bayesian Network Classifiers," *Machine learning*, vol. 29, no. 2-3, pp. 131-163, 1997.
- [30] Jiang, Liangxiao, Dianhong Wang, Zhihua Cai, and Xuesong Yan, "Survey of Improving Naive Bayes for Classification," *Paper presented at the International Conference on Advanced Data Mining and Applications*, 2007.

- [31] Bernard, Simon, Sébastien Adam, and Laurent Heutte, "Using Random Forests for Handwritten Digit Recognition," *Document Analysis and Recognition (ICDAR, 2007)*, vol. 2, pp. 1043-1047, 2007.
- [32] Dietterich, Thomas G., "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees," *Machine learning*, vol. 1, pp. 22, 1999.
- [33] Barandiaran, Iñigo, "The Random Subspace Method for Constructing Decision Forests," *IEEE Trans, Pattern Anal, Mach. Intell*, vol. 20, no. 8, pp. 1-22, 1998.
- [34] Kuncheva, Ludmila I., and Combining Pattern Classifiers, "Methods and Algorithms," John Wiley & Sons, New York, NY, 2004.
- [35] Breiman, Leo, "Random Forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [36] Marko Robnik, "Improving Random Forests," *ECML'04: Proceedings of the 15th European Conference on Machine Learning*, pp. 359-370, 2004.
- [37] Neves, Renata F. P., Cleber Zanchettin, and Alberto N.G. Lopes Filho, "An Efficient Way of Combining SVMs for Handwritten Digit Recognition," Paper presented at *the International Conference on Artificial Neural Networks*, pp. 229-237, 2012.
- [38] Owais M. Khanday, Samad Dadvandipour, "Convolutional Neural Networks and Impact of Filter Sizes on Image Classification," *Multidiszciplináris Tudományok*, vol. 10, no. 1, pp. 55-60, 2020.
- [39] M. Owais and S. Dadandipour, "The influence of the filter aggregates on the image classification using convolutional neural networks: a case study of handwritten digit recognition," in *Tavaszi Szél Konferencia 2019*, pp. 434, 2019.
- [40] Hassan, Ali, and Robert I. Damper, "Classification of Emotional Speech Using 3dec Hierarchical Classifier," *Speech Communication*, vol. 54, no. 7, pp. 903-916, 2012.

BIOGRAPHIES OF AUTHORS



Owais Mujtaba Khanday received his B.Sc. (I.T) from the University of Kashmir (S.P College Srinagar) and M.Sc. (Computer Science) from the University of Pondicherry. Currently, he is a Ph.D. Student at the University of Miskolc, Hungary, under the Stipendium Hungaricum Scholarship program.
aitowais@uni-miskolc.hu +36704202865



Dr. Samad Dadvandipour, Associate Professor, Institute of Information Sciences, University of Miskolc, 3515 Hungary
dr.samad@uni-miskolc.hu