

# Handwritten Digit Recognition by Support Vector Machine Optimized by Bat Algorithm

Eva Tuba

Faculty of Mathematics,  
University of Belgrade  
Studentski trg 16,  
11000 Belgrade, Serbia  
etuba@acm.org

Milan Tuba

Faculty of Computer Sci.,  
John Naisbitt University  
Bulevar umetnosti 29,  
11070 Belgrade, Serbia  
tuba@ieee.org

Dana Simian

Faculty of Science  
Lucian Blaga University,  
Ion Ratiu Street 5-7,  
550012, Sibiu, Romania  
dana.simian@ulbsibiu.ro

## ABSTRACT

Handwritten digit recognition is an important but very hard practical problem. This is a classification problem for which support vector machines are very successfully used. Determining optimal support vector machine is another hard optimization problem that involves tuning of the soft margin and kernel function parameters. For this optimization we adjusted recent swarm intelligence bat algorithm. We intentionally used weak set of features, four histogram projections, to prove that even under unfavorable conditions our algorithm would achieve acceptable results. We tested our approach on standard MNIST benchmark datasets and compared the results with other recent approaches from literature where our proposed algorithm achieved better results i.e. higher correct classification percentage.

## Keywords

Handwritten digit recognition, swarm intelligence, bat algorithm, support vector machine, parameter tuning.

## 1 INTRODUCTION

Nowadays many different applications need some object recognition and because of that it represents an active research field. Object recognition is a part of computer vision, which refers to the problem of recognition of specific object in digital image or digital video. Optical character recognition (OCR) is one subfield of object recognition while digit recognition is the widely studied part of OCR. Digit recognition is used in post offices for sorting the mail [NS12], in banks for reading checks [MAK10], for license plate recognition [CGA08], street number recognition [SCL12], etc.

Task of digit recognition can be divided into two groups, printed digit recognition and handwritten digit recognition. Recognition of printed digits is easier compared to the handwritten digit recognition because printed digits have regular shape and difference between images of the same number are just in the angle of view, size, color, etc. On the other hand, there are numerous handwriting styles which mean that

the same digit can be written in many different ways, hence more effort is required to find similarity between instances of the same digit.

One of the oldest techniques for object recognition is template matching. This technique is not suitable for handwritten digit recognition due to numerous variants in writing style, angle of writing, etc. In general, nowadays digit recognition contains three parts, preprocessing, feature extraction and classification. Preprocessing prepares image for feature extraction. Some of the common preprocessing steps are binarization, centering, morphological operations and more. Feature extraction is very important step and success of the classification strongly depends on it. Many different features were proposed in literature. In [JSDK13] horizontal and vertical projection with dynamic thresholding was proposed. Projection histograms are usually used for printed digit recognition and combined with other feature sets. Invariant moments such as geometric moments, affine invariant moments, Legendre moments, Zernike moments, Hu moments, etc. are the common choices for features [SSN16].

One of the most important parts of object recognition algorithms and handwritten digit recognition algorithms is classification. Classification in computer science represents prediction of class or label for an object based on its similarity with previous objects. In machine learning, each object or instance is represented with same set of features. Based on the learning

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

algorithm, classifiers can be divided to unsupervised and supervised classifiers. Supervised learning uses knowledge of labels for instances used for building the model while instances for unsupervised learning are unlabeled. Today, many techniques for building a classification model are used. One of the simplest machine learning algorithms is  $k$ -nearest neighbors (KNN). KNN is nonparametric technique that classifies instances by a majority vote of its neighbors. Instances will be assigned to the class most common amongst its  $K$  nearest neighbors measured by a distance function such as Euclidean distance, Manhattan distance, Hamming distance, etc. Decision tree represents rule based classifier widely used in different applications [ZWPJ14]. Some of the classifiers work with probability model and use statistical learning algorithms. These classifiers calculate probability that an instance belongs to each class. Linear combination of features that best describes difference between classes was found with these classifiers [CLY+11]. One of the recent proposed and widely used perception based classifier is artificial neural network (ANN). ANN is also used for prediction and pattern recognition [MLW+13b], [KPB08]. In the past few years one of the most used and most successful classifiers is support vector machine (SVM) [CV95]. Many applications use SVM for solving the classification problem, especially these for handwritten digit recognition. In [MUS08], SVM was used to improve classification accuracy for the OCR of mathematical documents. In [LMPS+15] it was used for classification of brain metastasis and radiation necrosis. In [GC04] support vector machines and neural network were combined for classification of handwritten digit recognition.

In this paper we propose using SVM for handwritten digits recognition. Support vector machine has a few parameters that should be adjusted. First parameter is parameter of soft margin  $C$  that allows outliers to be misclassified. In real life data, outliers are common and also data usually are not linearly separable. In that case some kernel functions need to be used and parameter of this functions also need to be tuned. One of the common kernel functions is Gaussian radial basis function with parameter  $\gamma$ . Tuning parameters of SVM is a hard optimization problem.

Bio-inspired algorithms such as swarm intelligence algorithms are widely researched and used for hard optimization problems. In swarm intelligence algorithms behavior of collectives of simple agents were simulated. Particle swarm optimization (PSO) is one of the oldest algorithm in this class of algorithms [KE95]. Today many different algorithms were proposed and used such as ant colony algorithm, artificial bee colony, cuckoo search, firefly algorithm and others.

Swarm intelligence algorithms have been used for SVM parameters tuning. In [BHX13] memetic algorithm based on PSO and pattern search was used for SVM parameters tuning. Modified PSO that uses chaotic mapping was introduced in [Wu11] for parameter optimization of SVM variant called wavelet  $v$ -support vector machine and in [LZ15] for improving classification accuracy of linear square SVM. In [MYK14] artificial bee colony was used for parameter tuning for linear square SVM. In [XBH14] firefly algorithm was used for optimization of multi-output support vector machine. A parallel time variant particle swarm optimization algorithm to simultaneously perform the parameter optimization and feature selection for SVM was proposed in [CYW+14].

In this paper we propose using SVM optimized by bat algorithm for handwritten digit recognition using intentionally weak features with which other approaches would not give good results. We propose usage of recent swarm intelligence algorithm, bat algorithm, for SVM parameter tuning. Our proposed algorithm was tested on standard MNIST [LBBH98] dataset for handwritten digit recognition and performance was better than other approaches from literature [MLW+13b], [KDB+13].

The rest of the paper is organized as follows. In Section 2 mathematical model for SVM is described. Section 3 then describes the bat algorithm. Explanation of our proposed algorithm for SVM parameter tuning is given in Section 4. Experimental results and comparison with other approaches from literature are given in Section 5. At the end conclusion along with proposed future work are presented in Section 6.

## 2 SUPPORT VECTOR MACHINE

Support vector machine was proposed by Vapnik as binary classifier [CV95]. It represents one of the latest supervised learning classifiers and it was used in numerous applications. SVM discovers a hyperplane that separates data from different classes. Each instance is labeled with one of existing classes and they are represented as points in space. SVM builds a model based on instances from training set and further classification of unknown instances is done by that model.

Hyperplane that separates labeled instances from the training set is defined by the next equation:

$$y_i(w \cdot x_i + b) \geq 1 \text{ for } 1 \leq i \leq n. \quad (1)$$

where  $x_i \in R^d$  are instances represented as vectors in  $d$ -dimensional space,  $n$  is the number of instances,  $y_i \in \{-1, 1\}$  are classes of corresponding instances and  $w$  and  $b$  are parameters of the hyperplane. This hyperplane is determined by the nearest instances that

are called support vectors. Hyperplane should be as far as possible from instances of both classes. The distance that should be maximized is  $\frac{2}{\|w\|}$ .

The described model has a problem to classify real life data, because all instances must be on the correct side of the hyperplane. Real world data contains some noise and usually a few outliers. The previous model is not able to separate such data. As a solution for this problem, using of soft margin was proposed. Soft margin is used instead of Eq. (1). The idea is to introduce a slack variable  $\varepsilon$  that allows some instances to be misclassified i.e. to be on the wrong side of the hyperplane. This is defined by the following expression:

$$y_i(w \cdot x_i + b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0, \quad 1 \leq i \leq n \quad (2)$$

Finding this hyperplane is done by solving the following quadric programming problem:

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \varepsilon_i, \quad (3)$$

where  $C$  is the soft margin parameter. Increasing the value of parameter  $C$  asymptotically leads to the model with hard margin. Selecting appropriate value for this parameter has major influence on classification accuracy [Wan05].

Another problem with this model, when it comes to real world data, is the assumption that instances are linearly separable. In order to make SVM suitable for non-linearly separable data kernel function is used instead of dot product. Theoretically, any function that satisfies Mercer's condition can be used as kernel function. In practice, usually Gaussian radial basis function (RBF), polynomial function and sigmoid function are used. Kernel function projects data into higher dimensional space in order to make it linearly separable. In this paper we used RBF as kernel function. RBF is defined by the next equation:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2). \quad (4)$$

where  $\gamma$  is the parameter of kernel function. This parameter has influence on the quality of classifier, so tuning the value of it is an important task. Too large value of  $\gamma$  will reduce benefits gained by introducing the kernel function and too small value will make decision boundary sensitive to the noise in training data.

Selecting optimal values for SVM's parameters is very important task. In [HCL10] most common technique for parameters tuning, grid search with cross validation, was described. Grid search builds models for different values of parameters and checks the accuracy of these models. Cross validation is used for determination of

the model's accuracy. Training set is divided into  $\nu$  distinct subsets. For training  $\nu - 1$  subsets were used and the accuracy was checked on the remaining subset. All subsets are used as test set once and the accuracy is the average value of  $\nu$  obtained accuracies. This method requires huge computational time and the search for optimal pair of values for  $(C, \gamma)$  is limited to predefined set of values.

Instead of the grid search, different stochastic optimization algorithms were successfully used [BHX13], [Wu11], [CYW+14], [MYK14].

### 3 BAT ALGORITHM

Bat algorithm is one of the recent swarm intelligence algorithm introduced by Yang [Yan10]. The algorithm was inspired with echolocation of bats which they use to detect pray and avoid obstacles. Bats emit sound pulses and navigate by using the time delay from emission to reflection.

Bat algorithm was widely used and studied in the past few years. In [YG12] it is used for multi-objective optimization and in [GYAT12] constrained optimization. In [HZL13] global engineering optimization and large-scale optimization problems were solved by bat algorithm. As a result of wide research of algorithm many improvements and hybridizations were developed [AT14], [FJFY13]. Numerous applications use bat algorithm for some real world hard optimization problems such as image processing [ZW12], RFID network planing [TB15], training neural networks [TAB15], etc.

Bat algorithm starts with initialization of the population that is performed randomly. Each bat from the population is represented by its location  $x_i^t$ , velocity  $v_i^t$ , frequency  $f_i^t$ , loudness  $A_i^t$  and the emission pulse rate  $r_i^t$  in a  $D$ -dimensional search space. Location and velocity are updated at each iteration based on the previous solution. The new solution is calculated according to the following equations [Yan10]:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (5)$$

$$v_i^t = v_i^{t-1} + (x_* - x_i^{t-1})f_i \quad (6)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (7)$$

where  $\beta$  is a random vector generated from uniform distribution from the closed range  $[0, 1]$  and  $x_*$  represents the current global best location which is found after comparing all the solutions among all the bats. At the beginning each bat is randomly assigned a frequency which is drawn uniformly from the interval  $[f_{min}, f_{max}]$ .

Every swarm intelligence algorithm has two important operations, exploration and exploitation. In the bat

algorithm for local search (exploitation) random walk with direct exploitation is used. It is defined by the following equation:

$$x_{new} = x_* + \varepsilon * A_t \quad (8)$$

where  $A_t = \langle A_i^t \rangle$  represents the average loudness of all bats at the time step  $t$  and  $\varepsilon$  is random number from the range  $[-1, 1]$ . Parameter  $\varepsilon$  defines intensity and direction of random walk. The local search depends on the rate  $r_i$  of pulse emission for the  $i$ -th bat. When bat approaches to its pray, bat becomes more silent, so the loudness decreases, but the pulse rate increases. For the purpose of the algorithm, the pulse rate can be defined in the range from 0 to 1, where 0 means that there is no emission at all and 1 means that the bat is emitting at their maximum [Yan10]. It can be formally represented by the next equations:

$$A_i^t = \alpha A_i^{t-1} \quad (9)$$

$$r_i^t = r_i^0 (1 - e^{-\gamma t}) \quad (10)$$

where  $\alpha$  and  $\gamma$  represent constants defined according to the problem that is solved by the bat algorithm.

Bat algorithm is summarized in Algorithm 1.

---

**Algorithm 1** Pseudo-code for the original bat algorithm

---

[Yan10]

Define the objective function  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)^T$

Initialize the population of bats  $x_i$ , ( $i = 1, 2, \dots, n$ ) i  $v_i$

Define pulse frequency  $f_i$  at the position  $x_i$

Initialize pulse rates emission  $r_i$  and sound loudness  $A_i$

**while**  $t < IN$  **do**

    Generate new solutions by adjusting frequency and updating velocities and locations solutions

    by using equations (5) - (7)

**if**  $rand > r_i$  **then**

        Select the best solution from the population

        Generate new solution in the neighborhood of chosen solution

**end if**

    Generate new solution by flying randomly (random walk)

**if**  $rand < A_i$  and  $f(x_i) < f(x_*)$  **then**

        Accept new solutions

        Increase  $r_i$  and decrease  $A_i$

**end if**

    Rank all the bats in the population and find the current best solution  $x_*$

**end while**

Post-process results and visualization

---

## 4 OUR PROPOSED ALGORITHM

In this paper we proposed using projection histograms as the feature set for handwritten digits. Projection histograms were usually used for typed digit recognition (e.g. license plate recognition). For

handwritten digits recognition projection histograms were not much used, especially without another set of features. We intentionally used this weak set of features to test our SVM classifier under unfavorable conditions. Fig. 1 shows example of projection histograms on  $x$ -axis for all 10 digits.



Figure 1: Projection histogram on  $x$ -axis ( $y = 0$ )

Because of various writing styles, thickness of pen, angle, etc., projection histograms can be very different for the same digit and on the other hand they can be very similar for different digits so projection on one axis cannot be sufficient. Fig. 2 shows example of histograms for digits 0, 3 and 8.

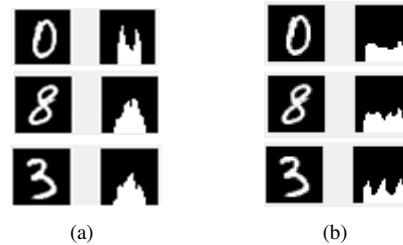


Figure 2: Example of histograms for numbers 0, 8 and 3 on (a)  $x$  and (b)  $y$  axis

It can be noticed that projection histograms on  $x$  axis for digits 8 and 3 are very similar, they have peak in the middle, but projections on  $y$  axis are different. Number 3 has three peaks, while number 8 has little dent in the middle. On the other hand projection histograms for digits 8 and 0 on  $y$  axis are similar, but difference is clear at projection histograms on  $x$  axis.

Besides these two projection histograms, in our algorithm we used two more projections, on lines  $y = x$  and  $y = -x$ , thus each digit was represented with four different projection histograms. Fig. 3 and Fig. 4 show examples of all four histograms for different samples of digit 3. It can be seen that projection histograms on one axis can be very different, but with the same characteristics, and combination of four histograms helps to differentiate between different digits.

Described feature set was used as input for support vector machine. For handwritten digit recognition, ten different classes are needed, one class for each digit. SVM is binary classifier while for this task multi-classification is needed. Two main techniques are used in cases like this. First, known as *one-against-all*, makes one model for each class. Each model separates one class from all others. This method is more suitable for classifiers that produce real valued probabilities that instance belongs to class. Second

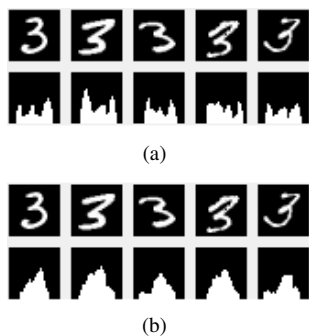


Figure 3: Number 3 histograms on (a)  $x$  and (b)  $y$  axis

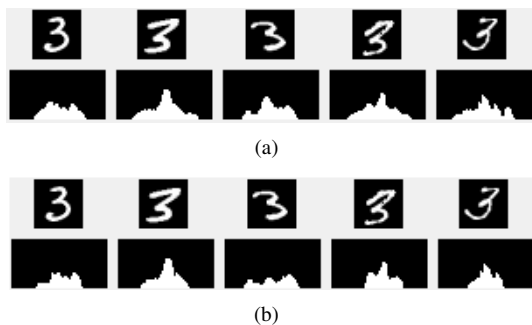


Figure 4: Number 3 histograms on (a)  $y = x$  and (b)  $y = -x$

method that is used for multi-classification with binary classifiers is *one-against-one*. If there are  $n$  classes then  $\frac{n(n-1)}{2}$  models need to be made, one model for each pair of classes. Class of an unknown instance can be determined by counting the votes. Each model produces result and class that was determined most times represents the class of unknown instance. In the case when two or more classes have the same number of votes, different methods can be used for making the final decision.

In this paper we proposed combination of the two mentioned techniques for multi-classification. Initially, classes were predicted by 10 different models (*one-against-all*). If the class was not determined uniquely or was not determined at all, we used *one-against-one* technique.

Important part of classification procedure is scaling. Feature values of training and test data should be scaled to range  $[0, 1]$  or  $[-1, 1]$ . Scaling values have significant influence on classification accuracy. Without scaling data in greater numerical range would dominate over data in smaller range. Also training and test data should be scaled with same factor.

Parameters of the SVM were tuned by bat algorithm. Dimension of search space was 2, search for optimal pair of values for  $C$  and  $\gamma$ . Objective function was to maximize accuracy of the SVM models. Accuracy was calculated with 10-fold cross validation as it was described in Section 2.

For different problems, parameters of the bat algorithm should be adjusted. Besides parameter adjustment, some other modifications may be needed. Pulse rate  $r$  and loudness  $A$  can be static for each bat or they can be changed according to Eq. 9 and Eq. 10. Speed of convergence of these two parameters is determined by the values of  $\alpha$  and  $\gamma$ . If pulse rate increases too fast, probability of random walk will be low. In order to ensure random walk, initial pulse rate should be closer to 1, so random walk would be performed in at least  $1 - r^0$  fraction of cases. Random walk will be performed even in the later cycles of the algorithm. Low values of loudness provide exploration. If loudness increases too fast it is possible to be trapped in local optima. Based on loudness, solution can be accepted even if it is not better than the current solution. This provides exploration and decreases the possibility of being trapped in local optimum. Loudness will increase with number of iteration, thus in later iterations in less cases generated solution would be accepted if it is not better. Another important parameter is frequency. Based on the range for frequency, new solution will be generated in some space around the global best position. For frequency value 1, new solution will be generated at the same point as the best solution. Large range for frequency allows wider space around the best solution for new solution. Depending on problem different frequency ranges should be used.

## 5 EXPERIMENTAL RESULTS

Quality of our proposed algorithm for handwritten digit recognition was tested on standard MNIST database [LBBH98]. In this database images were preprocessed so in this paper preprocessing was not included. All images were centered in a  $28 \times 28$  image. This database contains 60,000 images for training and 10,000 images for testing. We tested our algorithm on limited set of digits. Fig. 5 shows example of images from MNIST database.

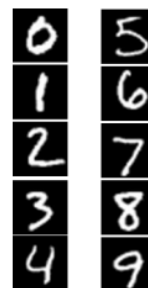


Figure 5: Example of digits from MNIST dataset

Proposed algorithm was implemented in Matlab R2015a and for classification LIBSVM (Version 3.21) [CL11] was used. Experiments were performed on the platform with the following features: Intel

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>0</b>	99	0	0	0	0	0	0	0	1	0
<b>1</b>	0	99	0	0	0	0	0	0	1	0
<b>2</b>	0	0	97	1	1	0	0	0	1	0
<b>3</b>	0	0	3	89	0	1	0	1	6	0
<b>4</b>	0	0	0	0	98	1	0	0	0	1
<b>5</b>	0	0	1	7	0	91	0	0	1	0
<b>6</b>	0	0	0	0	0	0	100	0	0	0
<b>7</b>	0	0	0	0	0	1	0	93	0	6
<b>8</b>	0	0	1	0	0	1	2	1	95	0
<b>9</b>	1	0	0	0	2	0	0	1	1	95

Table 1: Accuracy of classification for our proposed method (%)

® Core™i7-3770K CPU at 4GHz, 8GB RAM, Windows 10 Professional OS.

Parameters for bat algorithm were determined empirically, using theoretical insight from Section 4. Initial value of pulse rate was set to  $\gamma=0.92$  and loudness was  $\alpha=0.993$ . Frequency range was  $[-8, 10]$ .

Dimension of images are 28 by 28, so projection histograms on  $x$  and  $y$  axis contain 28 elements. Projection histograms on  $y = x$  and  $y = -x$  have 55 elements. Using combination of all four histograms as input vectors means that each instance contains 166 elements.

Search space for exponents of parameters of SVM were  $[0, 20]$  for  $C$  and for  $\gamma$  was  $[-15, 5]$  so the search space for parameters were  $[2^0, 2^{20}]$  and  $[2^{-15}, 2^5]$  since log scaling was used. Computational times were few minutes for each model but they are not very important since the model is constructed only once and offline.

Table 1 shows the results of classification. The best accuracy was achieved for digit 6 where all samples were recognized correctly. Drastically worse accuracy compared to other digits was achieved for recognition of digit 3. Digit 3 was recognized as 8 in 6% of cases and it was classified as 2 in 3% of cases. Only 89% of samples of digit 3 were recognized correctly.

Additionally, we compared our results with other state-of-the-art algorithms. In [KDB+13] use of multilayer neural network (MLNN) was proposed and dilatation algorithm combined with zoning techniques was used. Table 2 shows comparison of results reported in [KDB+13] and the results obtained with our proposed method.

In [KDB+13] the worst results were achieved for recognition of digit 9, while the best result was for digit 1. Our proposed model achieved significantly better results for each digit, thus the global accuracy was better as well. Our proposed feature set does not require any zoning technique so it is simpler to extract the features compared to [KDB+13] and the accuracy of classification is better.

Digit	MLNN	SVM-BAT
0	86.45	99.00
1	94.39	99.00
2	88.73	97.00
3	77.02	89.00
4	76.12	98.00
5	84.10	91.00
6	78.81	100.00
7	77.12	93.00
8	79.03	95.00
9	49.64	95.00
<b>Global</b>	79.14	<b>95.60</b>

Table 2: Accuracy of classification reported in [KDB+13] and our proposed method (%)

Another recent algorithm from literature was [MLW+13b] where three machine learning algorithms were proposed for handwritten digit recognition, extreme learning machine (ELM), regularized extreme learning machine (REML) and optimal weight learning machine (OWLM). Neural networks with different number of nodes were tested and compared. The best results were achieved by OWLM with 150 nodes. Man et al. in [MLW+13b] reported 85.16% as the best global accuracy, which is significantly less than accuracy of 95.60% achieved with our proposed method. With ELM learning algorithm and 150 nodes accuracy was 82.83% and with REML the highest accuracy that was achieved was 82.96%. Our proposed method produced better results compared to all results presented in [MLW+13b].

## 6 CONCLUSION

In this paper we proposed a novel algorithm for handwritten digit recognition. The goal was to use simple feature set as input for support vector machine that was used for classification. Optimal SVM models were determined by recent swarm intelligence algorithm, bat algorithm. Bat algorithm was adjusted and used for parameter tuning of the support vector machine. We tested our proposed method on standard

MNIST dataset and achieved global accuracy of 95.60%. We compared our method with other methods proposed in literature [KDB+13], [MLW+13b] and our proposed method obtained better accuracy with rather simple feature set. This establishes this approach as very robust and by using more complex features the results could be further improved. Additional validation can be done using other databases, for example USPS.

## 7 ACKNOWLEDGMENT

M. Tuba was supported by the Ministry of Education, Science and Technological Development of Republic of Serbia, Grant No. III-44006.

D. Simian was supported by the research grant LBUS-IRG-2015-01, project financed by Lucian Blaga University of Sibiu.

## 8 REFERENCES

- [AT14] Alihodzic, A. and Tuba, M. Improved hybridized bat algorithm for global numerical optimization. The 16th IEEE International Conference on Computer Modelling and Simulation, UKSim-AMSS 2014, pp. 57–62, 2014.
- [BGC12] Bhattacharya, G., Ghosh, K. and Chowdhury, A. S. An affinity-based new local distance function and similarity measure for kNN algorithm. *Pattern Recognition Letters*, 33(3):356–363, 2012.
- [BHX13] Bao, Y., Hu, Z., and Xiong, T. A PSO and pattern search based memetic algorithm for SVMs parameters optimization. *Neurocomputing*, 117:98–106, 2013.
- [CGA08] Caner, H., Gecim, H. S. and Alkar, A. Z. Efficient embedded neural-network-based license plate recognition system. *IEEE Transactions on Vehicular Technology*, 57(5):2675–2683, 2008.
- [CL11] Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, May 2011.
- [CLY+11] Chen, H.-L., Liu, D.-Y., Yang, B., Liu, J. and Wang, G. A new hybrid method based on local Fisher discriminant analysis and support vector machines for hepatitis disease diagnosis. *Expert Systems with Applications*, 38(9):11796–11803, 2011.
- [CV95] Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [CYW+14] Chen, H.-J., Yang B., Wang, S.-J., Wang G., Liu, D.-Y., Li, H., and Liu, W.-B. Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy. *Applied Mathematics and Computation*, 239:180–197, 2014.
- [FJFY13] Fister, I. Jr., Fister, D., and Yang, X.-S. A hybrid bat algorithm. *Elektrotehniski Vestnik/Electrotechnical Review*, 80(1-2):1–7, 2013.
- [GC04] Gorgevik, D. and Cakmakov, D. An efficient three-stage classifier for handwritten digit recognition. The 17th International Conference on Pattern Recognition (ICPR 2004), Volume 4, pp. 507–510, 2004.
- [GYAT12] Gandomi, A. H., Yang, X.-S., Alavi, A. H. and Talatahari, S. Bat algorithm for constrained optimization tasks. *Neural Computing and Applications*, 22(6):1239–1255, 2012.
- [HCL10] Hsu, C.-W., Chang, C.-C., and Lin, C.-J. A practical guide to support vector classification. Technical report, National Taiwan University, 2010.
- [HZL13] Huang, G.-Q., Zhao, W.-J. and Lu, Q.-Q. Bat algorithm with global convergence for solving large-scale optimization problem. *Application Research of Computers*, 30(5):1323–1328, 2013.
- [JSDK13] Jagannathan, J., Sherajdheen, A., Deepak, R. M. V. and Krishnan, N. License plate character segmentation using horizontal and vertical projection with dynamic thresholding. *International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN)*, pp. 700–705, 2013.
- [KDB+13] Kessab, B. E., Daoui, C., Bouikhalene, B. Fakir, M. and Moro, K. Extraction method of handwritten digit recognition tested on the MNIST database. *International Journal of Advanced Science and Technology*, 50(6):99–110, 2013.
- [KE95] Kennedy, J. and Eberhart, R. Particle swarm optimization. *IEEE International Conference on Neural Networks*, 1995., Volume 4, pp. 1942–1948, 1995.
- [KPB08] Kang, M. and Palmer-Brown, D. A modal learning adaptive function neural network applied to handwritten digit recognition. *Information Sciences*, 178(20):3802 – 3812, 2008.
- [LBBH98] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), pp. 2278–2324, 1998.
- [LMB+11] Lee, C.-C., Mower, E., Busso, C. Lee, S. and Narayanan, S. Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, 53(9-10):1162–1171, 2011.

- [LMPS+15] Larroza, A., Moratal, D., Paredes-Sanchez, A., Soria-Olivas, E., Chust, M. L., Arribas, L. A. and Arana, E. Support vector machine classification of brain metastasis and radiation necrosis based on texture analysis in MRI. *Journal of Magnetic Resonance Imaging*, 42(5):1362–1368, 2015.
- [LZ15] Liu, F. and Zhou, Z. A new data classification method based on chaotic particle swarm optimization and least square-support vector machine. *Chemometrics and Intelligent Laboratory Systems*, 147(October):147–156, 2015.
- [MAK10] Mahmoud, S. A. and Al-Khatib, W. G. Recognition of Arabic (Indian) bank check digits using log-Gabor filters. *Applied Intelligence*, 35(3):445–456, 2010.
- [MLW+13b] Man, Z., Lee, K., Wang, D., Cao, Z. and Khoo, S. An optimal weight learning machine for handwritten digit image recognition. *Signal Processing*, 93(6):1624–1638, 2013.
- [MUS08] Malon, C., Uchida, S. and Suzuki, M. Mathematical symbol recognition with support vector machines. *Pattern Recognition Letters*, 29(9):1326–1332, 2008.
- [MYK14] Mustaffa, Z., Yusof, Y. and Kamaruddin S. S. Enhanced artificial bee colony for training least squares support vector machines in commodity price forecasting. *Journal of Computational Science*, 5(2):196–205, 2014.
- [NS12] Niu, X.-X. and Suen, C.-Y. A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012.
- [SCL12] Sermanet, P., Chintala, S. and LeCun, Y. Convolutional neural networks applied to house numbers digit classification. *The 21st International Conference on Pattern Recognition (ICPR)*, pp. 3288–3291, 2012.
- [SSN16] Singh, P.-K., Sarkar, R. and Nasipuri, M. A study of moment based features on handwritten digit recognition. *Applied Computational Intelligence and Soft Computing*, 2016:1–17, 2016.
- [TAB15] Tuba, M., Alihodzic, A. and Bacanin, N. Cuckoo search and bat algorithm applied to training feed-forward neural networks. In: *Recent Advances in Swarm Intelligence and Evolutionary Computation*, volume 585 of *Studies in Computational Intelligence (Editor: Xin-She Yang)*, pp. 139–162. Springer International Publishing, 2015.
- [TB15] Tuba, M. and Bacanin, N. Hybridized bat algorithm for multi-objective radio frequency identification (RFID) network planning. *IEEE Congress on Evolutionary Computation (CEC2015)*, pp. 499–506, 2015.
- [Wan05] Wang, L. *Support Vector Machines: Theory and Applications*. Series: *Studies in Fuzziness and Soft Computing*, Volume 177, Springer-Verlag Berlin Heidelberg, 2005.
- [Wu11] Wu, Q. A self-adaptive embedded chaotic particle swarm optimization for parameters selection of Wv-SVM. *Expert Systems with Applications*, 38(1):184–192, 2011.
- [XBH14] Xiong, T., Bao, Y. and Hu, Z. Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting. *Knowledge-Based Systems*, 55(January):87–100, 2014.
- [Yan10] Yang, X.-S. A new metaheuristic bat-inspired algorithm. In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Volume 284 of the series *Studies in Computational Intelligence*: pp. 65–74, 2010.
- [YG12] Yang, X.-S. and Gandomi, A. H. Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, 29(5):464–483, 2012.
- [ZW12] Zhang, J.-W. and Wang, G. Image matching using a bat algorithm with mutation. *Applied Mechanics and Materials*, 203(1):88–93, 2012.
- [ZWPJ14] Zhang, Y., Wang, S., Phillips, P. and Ji, G. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowledge-Based Systems*, 64(July):22–31, 2014.