Bulletin of Electrical Engineering and Informatics
Vol. 10, No. 1, February 2021, pp. 290~298
ISSN: 2302-9285, DOI: 10.11591/eei.v10i1.2098

Selection of optimal hyper-parameter values of support vector machine for sentiment analysis tasks using nature-inspired optimization methods

Lakshmana Kumar Ramasamy[1], Seifedine Kadry[2], Sangsoon Lim[3]
[1]Department of Computer Applications, Hindusthan College of Engineering and Technology, Tamil Nadu, India
[2]Department of Mathematics and Computer Science, Faculty of Science, Beirut Arab University, Lebanon
[3]Department of Computer Engineering, Sungkyul University, Anyang, South Korea

Article Info

Article history:

Received Dec 26, 2019
Revised May 5, 2020
Accepted Jun 11, 2020

Keywords:

Heuristic optimization
Hyper-parameter selection
sentiment analysis
Nature-inspired method
Support vector machine

ABSTRACT

Sentiment analysis and classification task is used in recommender systems to analyze movie reviews, tweets, Facebook posts, online product reviews, blogs, discussion forums, and online comments in social networks. Usually, the classification is performed using supervised machine learning methods such as support vector machine (SVM) classifier, which have many distinct parameters. The selection of the values for these parameters can greatly influence the classification accuracy and can be addressed as an optimization problem. Here we analyze the use of three heuristics, nature-inspired optimization techniques, cuckoo search optimization (CSO), ant lion optimizer (ALO), and polar bear optimization (PBO), for parameter tuning of SVM models using various kernel functions. We validate our approach for the sentiment classification task of Twitter dataset. The results are compared using classification accuracy metric and the Nemenyi test.

Corresponding Author:

Sangsoon Lim,
Department of Computer Engineering,
Sungkyul University,
Anyang, South Korea.
Email: lssgood80@gmail.com

1. INTRODUCTION

Sentiment analysis [1-3] is a highly relevant research in the area of text analysis and mining. Many people post their views, opinion and ideas in unstructured format. The views are taken from the views of public, customer, social media, entertainment, sports, climate analysis and Industrial organization. Millions and billions of people and public are using social network websites such as Facebook, Twitter, and RenRen. The social media generates a huge volume of sentiment data in the various forms such as tweet id, status updates, reviews, author, content, tweets type and tweets status update. As the data size is going larger and larger, it is necessary to analyze and categorize the sentiment reviews or opinion of the various people to predict.

Machine learning techniques such as support vector machine (SVM) [4] is one of the frequently used techniques in sentiment data analysis to classify the tweets or author comments in the form of positive valence, negative valence and neutral based on the tweet data. For example, Silva *et al*. [5] developed a method which classifies the sentiment of twitter data with the use of lexicons and classifier groups. For classification, SVM, Naive Bayes (NB), random forest (RF), multinomial Naive Bayes (MNB) and Logistic Regression are used. Medhat *et al*. [6] survey the recent techniques used in analysis of sentiments and 54 articles were classified and summarized. For sentiment classification tasks, SVM and Naive Bayes

classifiers were commonly used. Bifet *et al.* [7] analyzed sentiment classification of Twitter messages using MNB, stochastic gradient descent and Hoeffding tree, and proposed sliding window-based kappa value statistics to evaluate the time changing based data streams.

Zainuddin *et al*. [8] created a method to classify the twitter-based sentiment using principal component analsysis (PCA). This is combined with sentiwordnet lexicon-based method and SVM classification. Liao [9] proposed a sophisticated artificial neural network (ANN) approach for analyzing Twitter data to perform sentiment analysis. Tripathy *et al.* [10] compares naive bayes and SVM on polarity movie dataset. SongboTan [11] performed sentiment classification on Chinese documents by applying the numerous features or attribute selection techniques such as information gain (IG), mutual information (MI), chi-square, and k-nearest neighbor (KNN), SVM, NB, and centroid classifiers. Abbasi [12] used weighted genetic algorithm (EWGA) and SVM with the benchmark movie dataset. Go *et al*. [13] explored SVM, Maximum Entropy, NB, achieving the accuracy of 80% using tweets with emoticon data. Arimaki [14] investigated stacked SVM based classification techniques to categorize the sentiments of Facebook posts. Kapočiūtė-Dzikienė *et al*. [15] compared traditional machine learning approaches (including SVM) and deep learning methods and found out that traditional methods still outperform neural networks for sentiment analysis tasks.

The problem of selecting the optimal values of SVM classifier parameters (or hyper-parameters) has been addressed by multiple researchers [16-18]. Damaševičius [19] adopted the nelder-mead (downhill simplex) algorithm for selection of SVM hyper-parameters for linear, polynomial and power series kernels. Miranda *et al.* [20] suggested a hybrid multi-objective optimization architecture, which aggregated meta-learning (ML) with particle swarm optimization (PSO) algorithm technique to refine a Pareto front of SVM configurations based on previous learning problems. Chang and Chou [21] perform maximization of a hyper-plane margin-based criterion. Next, the L2-soft-margin parameter C is obtained by a jackknife estimate of the eigenvalue perturbation of the SVM kernel matrix. Czarnecki *et al*. [22] used the Bayesian and random search optimization of SVM hyper-parameters.

Lin *et al.* [23] proposed the modified artificial fish swarm algorithm (AFSA) that employs the intelligence of fish swarms to enhance feature selection and parameter optimization for SVM models. Lin *et al.* [24] used a modified cat swarm optimization (CSO), a meta-heuristic based on the the behavior of cats, for improving search efficiency within the problem space of parameter values of SVMs. Cho and Hoang [25] employed a particle swarm optimization (PSO) to select appropriate input features and optimize SVM parameters in order to increase the accuracy of classification. Ji *et al*. [26] used ensemble Kalman filter (EnKF), an iterative optimization technique, for the SVM hyper-parameter tuning problem. Qin *et al.* [27] employ the chaotic PSO algorithm, in which chaotic sequences solve the premature convergence problem and boost the performance of PSO. Tharwat *et al.* [28] suggested the bat algorithm (BA) to optimize the parameters of SVM. Rojas-Dominguez *et al.* [29] explored and compared a variety of nature-inspired heuristics, namely firefly algorithm, BA, PSO, fruit fly algorithm, univariate marginal distribution algorithm (UMDA), and Boltzmann-UMDA, for proper tuning of SVM hyper-parameters. Hoang *et al.* [30] propose differential PSO-based technique to optimize the SVM parameters. Candelieri *et al*. [31] proposed a parallel global optimization model for tuning the SVM regression hyper-parameters.

The aim of this study is to apply and compare three heuristics, nature-inspired optimization techniques, cuckoo search optimization, ant lion optimization, and polar bear optimization, for finding optimal values of SVM hyper-parameters for sentiment classification tasks. The paper provides novel results since, for our knowledge, these three algorithms have not been applied and compared for SVM hyper-parameter tuning before.

The remaining parts of the paper are organized as follows. Section 2 describes the principles of SVM and the SCO, ALO, and PBO algorithms. Section 3 presents the experimental results on sentiment classification in Twitter dataset. Finally, section 4 presents the conclusions.

2. RESEARCH METHOD

2.1. Support vector machine

Support vector machine (SVM) is a supervised machine learning method. SVM uses training data to separate and construct a maximum margin hyperplane that can be used for classification. It is defined as follows:

$$f(x) = w \cdot x + b, w \in R^d, b \in R \quad (1)$$

where d is the dimensionality of data space, w is the weight factor, b denotes bias of the function, and x is the training data vector.

The optimal hyperplane of SVM is defined as:

$$\|w \cdot x - b\| = 0 \tag{2}$$

Finding an optimal hyperplane can be formulated as an optimization problem:

$$\min \|w\|, subject \quad y^T \left(w \cdot x - b\right) \geq 1 \tag{3}$$

where y is a vector of class labels (positive or negative).
Then the process of classification is a function:

$$y \mapsto \text{sgn}\left(w \cdot x - b\right) \tag{4}$$

The SVM kernel is a function *k* defined as a dot product of data inputs X_i and X_j:

$$k\left(X_i^T, X_j\right) = \varphi\left(X_i^T\right) \cdot \varphi\left(X_j\right) \tag{5}$$

where φ is some linear or non-linear data transform.

The commonly used kernel functions are linear, polynomial, RBF and sigmoid. Here we also use the power series kernel defined in [19]. The kernel functions and their hyper-parameters are detailed in Table 1. An important issue of SVM classifier is the selection of various hyper-parameters. The SVM hyper-parameters are C is a leverage between the training error and hyperplane margin, C>0; Q is largest size of quadratic programming (QP) sub-problems for SVM optimization, Q≥2; J is cost-factor by which training errors on positive instances outbalance errors on negative instances, J≥1.

Table 1. Support vector machine (SVM) kernels and their hyper-parameters

SVM kernel	Equation	Hyper-parameters
Linear	$X_i^T \cdot X_j$	C
Radial basis function (RBF)	$\exp\left(-\gamma \left\|X_i - X_j\right\|^2\right)$	C, γ
Sigmoid	$\tanh\left(\gamma \cdot X_i^T \cdot X_j + r\right)$	C, γ, r
Polynomial	$\left(\gamma \cdot X_i^T \cdot X_j + r\right)^d$	C, γ, r, d
Power series	$\sum_{k=1}^{n} a_k \left(X_i^T \cdot X_j - r\right)^k$	$a_1 ... a_n, r$

2.2 Cuckoo search optimization

Cuckoo search optimization (CSO) is an optimization algorithm proposed in [32]. It was motivated by the behaviour of some cuckoo species to lay their eggs in the nests of other hosts. Here we use a modification of CSO described in [33]. CSO uses an aggregation of local and global random walk, where each step value is selected from Lévy distribution for Lévy flights as follows:

$$x_{t+1} = x_t + a \cdot Levy\left(s, \lambda\right) \tag{6}$$

where $x_t \in ¤$ is a current solution in parameter space ¤, x_{t+1} is the next solution, a is the scaling factor, s is the step size, and λ is the parameter of Levy distribution. The CSO algorithm is described in pseudo-code in Figure 1.

```
Algorithm: Cuckoo Search Optimization
   Generate iteration time
   Initialize with random vector values and parameters
   Evaluate the fitness of each individual
   Find the best individual with the best fitness
   Iteration count t = 0
   while (stopping criterion is not met or t < MaxIterationCount)
     Get a random cuckoo m local random walk or Lévy Flight
     Evaluate its fitness F(m)
     Choose a nest n randomly
     if (F(m) < F(n))
       replace nest n by nest m
     end if
     Abandon a fraction of worse nests and build new nests
     Keep best nests with quality solutions
     Rank solutions and find best_solution
     Increase iteration count
   end while
return best-solution
```

Figure 1. Algorithm of cuckoo search optimization

2.3. Ant lion optimization

Ant lion optimizer (ALO) is an optimization algorithm proposed in [34]. It was inspired by the interaction of ants and ant-lions in nature. Here we use a modification of ALO described in [35]. In ALO, the position of ants is updated using the random walk equation:

$$x^{i}_{t+1} = \frac{\left(x^{i}_{t} - a_i\right) \times \left(d_i - c_i^t\right)}{\left(d_i^t - a_i\right)} + c_i \tag{7}$$

where a_i is the smallest walk of i-th variable, d_i is the largest walk in i-th variable, c_i^t is the smallest value of i-th variable at t-th iteration, and d_i^t indicates the largest value of i-th variable at t-th iteration. The ALO algorithm is described in pseudo-code in Figure 2.

```
Algorithm: AntLion Optimization
Initialize randomly the initial population of ants and antlions
Calculate the fitness of ants and antlions
Find the best antlions
while the stopping criterion is not satisfied
   for each ant
     Select an antlion using the Roulette wheel
     Update c and d values
     Create a random walk
     Update the position of ant
   end for
   Calculate the fitness of all ants
   Replace an antlion with its corresponding ant it if becomes fitter
   Update best_solution if antlion becomes fitter than the best_solution
end while
return best-solution
```

Figure 2. Algorithm of ant lion optimization

2.4. Polar bear optimization

Polar bear optimization (PBO) is an optimization algorithm proposed in [36]. It was inspired by the hunting and reproduction behaviour of polar bears in arctic conditions:

$$\left\| w \cdot x - b \right\| = 0 \tag{2}$$

$$x^{i}_{t+1} = \frac{\left(x^{i}_{t}\right)^{(best)} + \left(x^{i}_{t}\right)^{(10\%)}}{2} \quad (8)$$

where $\left(x^{i}_{t}\right)^{(best)}$ is the best individual and $\left(x^{i}_{t}\right)^{(10\%)}$ is the individuals ranked among best 10%. The PBO algorithm is described in pseudo-code in Figure 3.

```
Algorithm: Polar Bear Optimization
Define fitness function f, size of space solution a?, b?, maximum iteration count t,
maximum size of population n, maximum distance of vision θ, Generate a population
consisting of 75%n bears at random,
Assign iteration i:=0
While i ? T do
 for each polar bear (x¯t)in population do
    Find all angle values? at random
    Calculate radius r and new position (x¯t)new
  if f((x¯t)new)<f(x¯t)then
    Move the bear (x¯t)actual=(x¯t)new,
  else
   Calculate new position of the bear (x¯t)
  if f((x¯t)new)<f((x¯t)actual) then
   Move the bear (x¯t)actual=(x¯t)new,
  end if
 end if
end for
Randomly select one of the top 10% of bears,
Calculate the new position,
if f((x¯t)new)<f((x¯t)actual) then
Move the bear (x¯t)actual=(x¯t)new,
end if
Sort population according to the fitness function,
Choose value k € 0,1?
if (i<t? 1 and k>0.75) then
Choose two bears from the top 10% of population and add a reproduced bear
else if (number of bears>0.5 and k<0.25) then
remove the worst individual in the population
end if
i++,
end while
return the fittest pdar bear (x¯t)best
```

Figure 3. Algorithm of polar bear optimization

2.5. Evaluation

The performance of classifiers is evaluated using the accuracy metric, which calculates the ratio of true hits to the number of total guesses. To compare the solutions, following the suggestion of Wainer and Cawley [37], we used the multi-classifier/multi-data set procedure proposed by Demšar [38].

3. RESULTS AND DISCUSSION

3.1. Dataset

We used the publicly available Twitter dataset. Tweet is a user's opinion that is expressed emotionally by different people. The twitter dataset used in this work is labeled into three classes, i.e. positive, negative, and neutral. The positive emotion sentiments are surprise, love, affection, happiness, joy, smile etc., which refer to positive thinking nature of the person, create a happy environment and good for individual as well as for the society. Negative sentiments mean sadness, worry, jealous, and hate etc., and reflect the negative psychological state of the individual. The following pre-processing was performed to clean the data: removed all punctuations, @, _ symbols, numbers, and sequence of repeated characters; replaced all the emoticons with their sentiment value; removed stop words, and unnecessary white spaces.

The 1000 tweets were taken for analysis and split into 70% training set and 30% testing set. As a result, our dataset has 701 training samples and 299 testing samples with four predictors and three classes with 578 negative, 247 neutrals and 175 positive instances. All computations were done using MATLAB (MathWorks Inc, Natick, MA, USA) version 8.1.0.604 (R2013a).

3.2. Results

The 10-fold cross-validation procedure was performed for three times and the accuracy values were calculated. The procedure was repeated for each of five SVM kernels (linear, RBF, sigmoid, polynomial (cubic), and power series (n=5)) described in Table 1. The hyper-parameters of each kernel were optimized using three nature-inspired optimization algorithms (CSO, ALO, and BPO). The results of sentiment classification are summarized in Figures 4 and 5. When comparing the results by hyper-parameter optimization methods, the methods performed fairly similarly: CSO achieved the average accuracy of 0.8686, followed by ALO-0.8626 and PBO-0.8444. By kernel, the best average accuracy was achieved by power series (n=5) kernel-0.9088, followed by sigmoid-0.8780, cubic polynomial (0.8604), RBF (0.8385), and linear (0.8070) kernels.

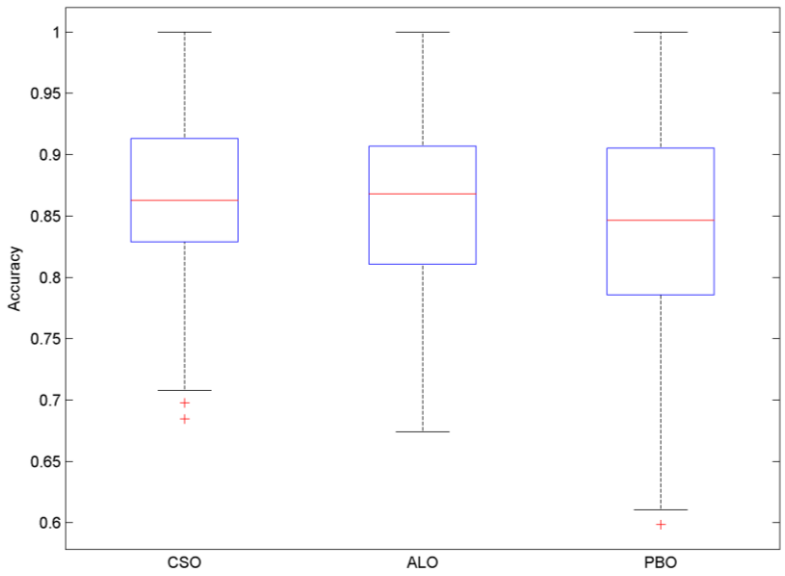

Figure 4. Comparison of accuracy of SVM classifier models by optimization method

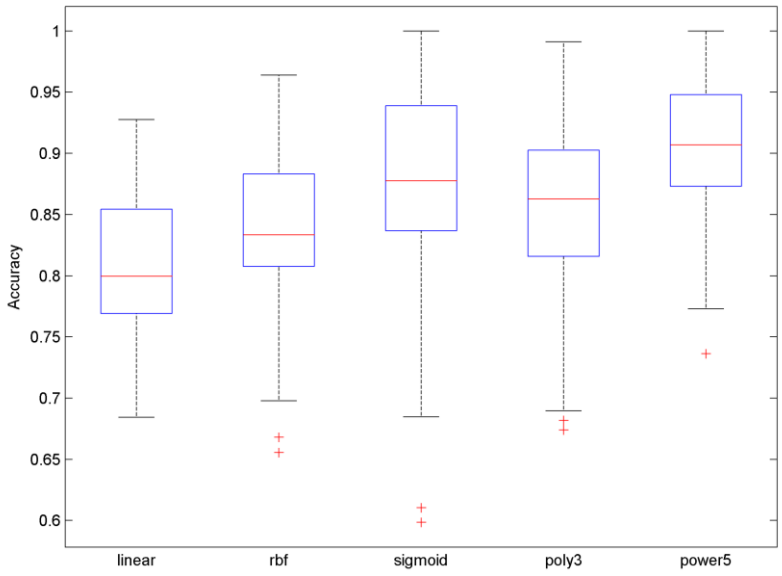

Figure 5. Comparison of accuracy of SVM classifier models by kernel function

To compare the hyper-parameter optimization solutions against each other, we performed the one-way analysis of variance (ANOVA) test with the following results: $F(14,435) = 14.4$, $p=10^{-28}$. Then

we performed the post-hoc Nemenyi test for each of groups (by optimization method, by kernel function as well as by combinations of optimization method and kernel function), which included Friedman test. The resulting p-values of the Friedman test were all below *0.05* (see in Table 2), thus reject the hypothesis that all optimization methods and SVM kernel functions are equivalent. The ranks of the optimization methods and kernel functions by sentiment classification accuracy are visualized using Demšar plots in Figures 6 and 7. The ranking results indicate the CSO and ALO are equivalent, but significantly better than BPO. On the other hand, kernel functions power series (n=5) and sigmoid are equivalent, but significantly better performing than other SVM kernel functions. When considering all combinations of optimization methods and kernel functions, the combination of CSO and power series (n=5) kernel performed the best, but not significantly better than ALO and power series (n=5) kernel as well as ALO and sigmoid kernel.

Table 2. P-values and critical distances of Friedman test

Analysis	P-value	Critical distance
By optimization method	0.005	0.27
By SVM kernel function	$1 \cdot 10^{-19}$	0.64
Combined	$1 \cdot 10^{-24}$	3.91

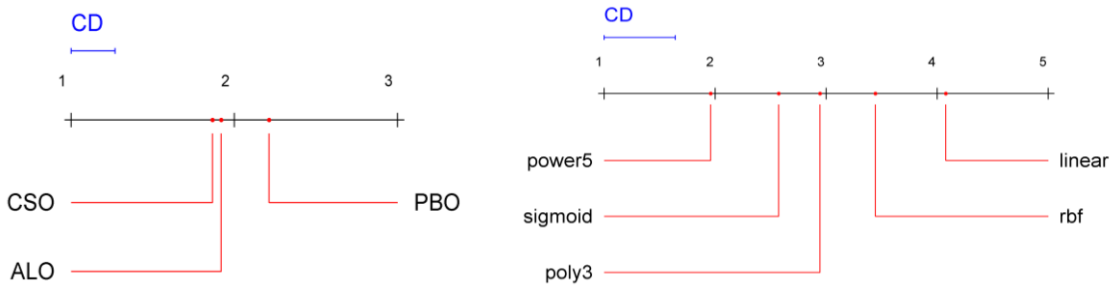

Figure 6. Demšar plot comparing SVM hyper-parameter optimization methods (left) and SVM kernel functions (right) following the Nemenyi test

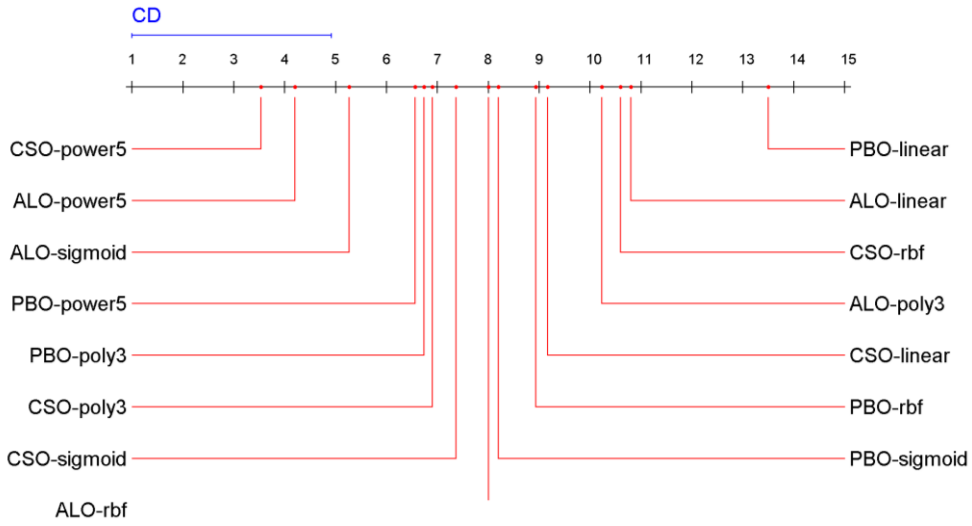

Figure 7. Demšar plot comparing all combinations of SVM hyper-parameter optimization methods and SVM kernel functions following the Nemenyi test

4. CONCLUSION

In this work, the analyzed the application of three heuristics, nature-inspired optimization techniques, cuckoo search optimization (CSO), ant lion optimizer (ALO), and polar bear optimization (PBO), for parameter tuning of SVM models using various kernel functions (linear, radial basis function, sigmoid, polynomial (cubic), and power series (n = 5)). We validated our approach for the sentiment classification task

of Twitter dataset. The results were compared using classification accuracy and statistical testing. The results show that the combination of CSO and ALO methods with power series (n=5) and sigmoid kernels outperformed the BPO method and other SVM kernel functions in terms of the classification accuracy. Future work will focus on extending our experiments to other classification tasks and datasets.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2018R1C1B5038818).

REFERENCES

[1] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," *Expert Systems with Applications*, vol. 40, no. 2, pp. 621-633, 2013.
[2] A. Ortigosa, J. M. Martín, and R. M. Carro, "Sentiment analysis in Facebook and its application to e-learning," *Computers in Human Behavior*, vol. 31, pp. 527-541, 2014.
[3] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10760-10773, 2009.
[4] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.
[5] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka Jr. "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, pp. 170-179, 2014.
[6] W. Medhat, A. Hassan, and H. Korashy "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093-1113, 2014.
[7] A. Bifet and E. Frank, "Sentiment knowledge discovery in twitter streaming data," *International Conference on Discovery Science*, pp. 1-15, 2010.
[8] N. Zainuddin, A. Selamat, and R. Ibrahim, "Twitter feature selection and classification using support vector machine for aspect-based sentiment analysis," *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 269-279, 2016.
[9] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, "CNN for situations understanding based on sentiment analysis of twitter data," *Procedia Computer Science*, vol. 111, pp. 376-381, 2017.
[10] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentimental reviews using machine learning techniques," *Procedia Computer Science*, vol. 57, pp. 821-829, 2015.
[11] S. Tan and J. Zhang, "An empirical study of sentiment analysis for Chinese documents," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2622-2629, 2008.
[12] A. Abbasi, A. Salem, and H. Chen, "Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums," *ACM Transactions on Information Systems*, vol. 26, no. 3, pp. 1-34, 2008.
[13] A. Go, R Bhayani, and L Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report*, Stanford, vol. 1, no. 12, 2009.
[14] E. Aramaki, S. Maskawa, and M. Morita, "Twitter catches the flu: Detecting influenza epidemics using Twitter," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1568-1576, 2011.
[15] J. Kapočiūtė-Dzikienė, R. Damaševičius, and M. Woźniak, "Sentiment analysis of Lithuanian texts using traditional and deep learning approaches," *Computers*, vol. 8, no. 1, p. 4, 2019.
[16] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 131-159, 2002.
[17] C-L. Huang and C-J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231-240, 2006.
[18] S-W. Lin, K-C. Ying, S-C. Chen, and Z-J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines." *Expert Systems with Applications*, 35, no. 4, 1817-1824, 2008.
[19] R. Damaševičius, "Optimization of SVM parameters for recognition of regulatory DNA sequences," *TOP*, vol. 18, no. 2, pp. 339-353, 2010.
[20] P. B. C. Miranda, R. B. C. Prudêncio, A. P. L. F. de Carvalho, and C. Soares, "A hybrid meta-learning architecture for multi-objective optimization of SVM parameters," *Neurocomputing*, vol. 143, pp. 27-43, 2014.
[21] C-C. Chang, and S-H. Chou, "Tuning of the hyperparameters for L2-loss SVMs with the RBF kernel by the maximum-margin principle and the jackknife technique," *Pattern Recognition*, vol. 48, no. 12, pp. 3983-3992, 2015.
[22] W. M. Czarnecki, S. Podlewska, and A. J. Bojarski, "Robust optimization of SVM hyperparameters in the classification of bioactive compounds," *Journal of Cheminformatics*, vol. 7, no. 1, p. 38, 2015.
[23] K-C. Lin, S-Y. Chen, and J. C. Hung, "Feature selection and parameter optimization of support vector machines based on modified artificial fish swarm algorithms," *Mathematical Problems in Eng.*, vol. 2015, p. 604108, 2015.
[24] K-C. Lin, Y-H. Huang, J. C. Hung, and Y-T. Lin, "Feature selection and parameter optimization of support vector machines based on modified cat swarm optimization," *International Journal of Distributed Sensor Networks*, vol. 11, no. 7, p. 365869, 2015.
[25] M-Y. Cho and T. T. Hoang, "Feature selection and parameters optimization of SVM using particle swarm optimization for fault classification in power distribution systems," *Computational Intelligence and Neuroscience*, vol. 2017, p. 4135465, 2017.

[26] Y. Ji, Y. Chen, H. Fu, and G. Yang, "An EnKF-based scheme to optimize hyper-parameters and features for SVM classifier," *Pattern Recognition*, vol. 62, pp. 202-213, 2017.
[27] C. Qin, Z. Xue, Q. Feng, and X. Huang, "Selecting parameters of an improved doubly regularized support vector machine based on chaotic particle swarm optimization algorithm. *Journal of Universal Computer Science*, vol. 23, no. 7, pp. 603-618, 2017.
[28] A. Tharwat, A. E. Hassanien, and B. E. Elnaghi, "A BA-based algorithm for parameter optimization of support vector machine," *Pattern Recognition Letters*, vol. 93, no. 13-22, 2017.
[29] A. Rojas-Domínguez, L. C. Padierna, J. M. Carpio Valadez, H. J. Puga-Soberanes, and H. J. Fraire, "Optimal hyper-parameter tuning of SVM classifiers with application to medical diagnosis," *IEEE Access*, vol. 6, pp. 7164-7176, 2018.
[30] T. T. Hoang, M-Y. Cho, M. N. Alam, and Q. T. Vu, "A novel differential particle swarm optimization for parameter selection of support vector machines for monitoring metal-oxide surge arrester conditions," *Swarm and Evolutionary Computation*, vol. 38, pp. 120-126, 2018.
[31] A. Candelieri, I. Giordani, F. Archetti, K. Barkalov, I. Meyerov, A. Polovinkin, and N. Zolotykh, "Tuning hyperparameters of a SVM-based water demand forecasting system through parallel global optimization," *Computers and Operations Research*, vol. 106, pp. 202-209, 2019.
[32] X. Yang and Suash Deb, "Cuckoo search via Lévy flights," *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210-214, 2009.
[33] M. Woźniak, D. Połap, C. Napoli, and E. Tramontana, "Graphic object feature extraction system based on cuckoo search algorithm," *Expert Systems with Applications*, vol. 66, pp. 20-31, 2016.
[34] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80-98, 2015.
[35] K. Książek, D. Połap, M. WoZniak, and R. Damaševičius, "Radiation heat transfer optimization by the use of modified ant lion optimizer," *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1-7, 2017.
[36] D. Polap and M. Woźniak, "Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism," *Symmetry*, vol. 9, no. 10, p. 203, 2017.
[37] J. Wainer and G. Cawley, "Empirical evaluation of resampling procedures for optimising SVM hyperparameters," Journal of Machine Learning Research, vol. 18, no. 15, pp. 1-35, 2017.
[38] A. Madi, O. Kassem Zein, S. Kadry, "On the improvement of cyclomatic complexity metric," *International Journal of Software Engineering and Its Applications*, vol. 7, issue 2, pp. 67-82, 2013.