SVM Hyper-Parameter Optimization for Sentiment Analysis Using Termite Alate Optimization Algorithm

Faiza Merah[1,*,†], Nawel Falek[2] and Hichem Houassi[3]

[1,2]*University Abbes Laghrour - Khenchela, Algeria*
[3]*ICOSI Laboratory, University Abbes Laghrour - Khenchela, Algeria*

Abstract
Support vector machine is a machine learning algorithm that is widely used in sentiment analysis, but they rely on a set of hyper-parameters which greatly influence their performance. Fine-tuning these hyper-parameters is a complex problem that needs experience and domain knowledge. Thus, this study considers it as a combinatorial optimization problem and a swarm-based optimization technique is used to tune the SVM hyper parameters in sentiment analysis by using the recently introduced termite alate optimization algorithm. The performance of the proposed approach is evaluated using five metrics: accuracy, precision, recall, F1-measure and computation time on five well known datasets in the field of sentiment analysis. The experimental results result in significant improvement in the SVM performance with optimized hyper-parameters by the proposed approach compared to SVM with default parameters in all datasets.

Keywords
Sentiment Analysis, Support Vector Machine, Hyper-parameter Tuning, Termite Alate Optimization Algorithm

1. Introduction

In recent decades, user-generated content on the Web and social media has experienced an extraordinary explosion. This has led to significant new challenges for companies, which are giving more attention to the content of communities on the Web in order to follow their various trends. An important element of such analysis is to characterize the sentiment expressed in comments on a specific topic, which is called sentiment analysis [1].

Sentiment Analysis (SA), also known as Opinion Mining, is a rapidly growing field with numerous applications. It involves assessing whether a piece of text is positive, negative, or neutral, and can be applied to various types of content such as reviews, articles, and social media posts. Supervised machine learning methods are commonly utilized to detect and classify user opinions on the Internet [2]. It has a wide range of algorithms including artificial neural networks, Support Vector Machines (SVM), random forests, naive Bayes, and K-nearest neighbors. SVM is a set of supervised learning techniques intended to solve classification problems. SVM is a generalization of linear classifiers which is quickly adopted for its ability to work with large data, the low number of hyper-parameters and their good results [3].

The performance of SVM hinges significantly on the meticulous selection of parameters. In the SA field, SVM is used generally by their default parameters [4], because the selection of appropriate hyper-parameters (HPs) values is a complex process. For this, we consider in this work the hyper-parameters tuning as an optimization problem and the new swarm-based optimization approach Termite Alate Optimization Algorithm (TAOA) is adapted and used to tune the SVM's hyper-parameters for SA. The TAOA is a recently proposed optimization algorithm by Arindam Majumder [5] for solving the optimization problems. The TAOA shows promising results compared to other swarm-based optimization algorithms in optimization of mathematical functions [5]. Given these promising results, in this paper we used TAOA as an optimization algorithm to tune the SVM's HPs in the SA field. The rest of this paper is organized as follows: Section two presents the background of the work. In the third Section we present the related work of the hyper-parameters tuning. In Section four, the proposed

The 13th Conference on Research in computIng at Feminine, May 20-21, 2024, Constantine, Algeria
*Corresponding author.
faiza.merah20@gmail.com (F. Merah)

approach is detailed. The section five presents experimental results and discussion. And finally, we conclude this paper in Section six by giving some perspectives and future works.

2. Related work

SVM's HPs optimization is a process of finding the best combination of HPs's values that optimize the performance of the SVM model. Several methods are introduced in the literature which are classified into Grid Search (GS) and Random Search (RS). Grid search, as a traditional method for HPs optimization [6], systematically explores the HPs space by generating a cartesian product of all possible combinations. It then calculates the performance score for each model to evaluate them. GA suffers from limitations, notably slow convergence and high dimensionality. The complexity of GS grows exponentially with the number of parameters (k) and distinct values (n) tested [7].

RS methods offer an alternative to GS by sampling the HPs space and evaluating sets from a specified probability distribution, RS randomly selects HPs sets to assess their performance. However, in RS methods the complexity scales linearly with the number of evaluations [8]. Additionally, it lacks a strategy for predicting the next trial, unlike some other optimization techniques [9]. Most works in HPs tuning tends to focus on GS and RS, or a comparison between them [10]. The authors in these researches framed the tuning problem in statistical terms and proposed metrics to quantify the tunability of algorithms' HPs.

In [11] the authors demonstrated significant enhancements in SVM performance through parameter optimization. They employed two methods: GS and Genetic Algorithm (GA). Comparative analysis revealed that GA outperformed GS. As described in [12], the authors advocated for employing an evolutionary algorithm called SHADE to optimize the deep learning model tailored for SA of Spanish tweets. Their findings illustrated that the HPs identified by the SHADE notably improved the performance of the deep learning approach.

3. Background

Here's a brief description of Support Vector Machines and the Termite Alate algorithm:

3.1. Support vector machine (SVM)

Support vector machine is a machine learning method that employs the principle of Structural Risk Minimization to identify the optimal hyperplane for distinguishing between two classes within the input space [13]. SVM is highly efficient in addressing challenges associated with textual data due to its ability to handle high-dimensional datasets [14], and effectively deal with correlated features by creating linear boundaries between distinct categories [15]. However, a kernel function does used to resolve the linearly inseparable problem by mapping it to linearly separable problem [16]. The objective is to find a hyperplane according to Equation 1 that optimally separates the data while Figure 1 shows that there are several valid hyperplanes for classification, all capable of separating the data set into different classes.

$$\mathrm{f}(x) = wx + b = 0 \tag{1}$$

Where w is the weight vector (normal to the plane) and b is the bias term.

Before turning to SVM hyper-parameters, let's explore the difference between hyper-parameters and parameters.

The terms "parameter" and "hyper-parameter" are often used in machine learning to describe two different types of variable. Parameters are the internal variables of the model that are automatically adjusted during training. They are learned from the training data and represent the weights or coefficients of the model.

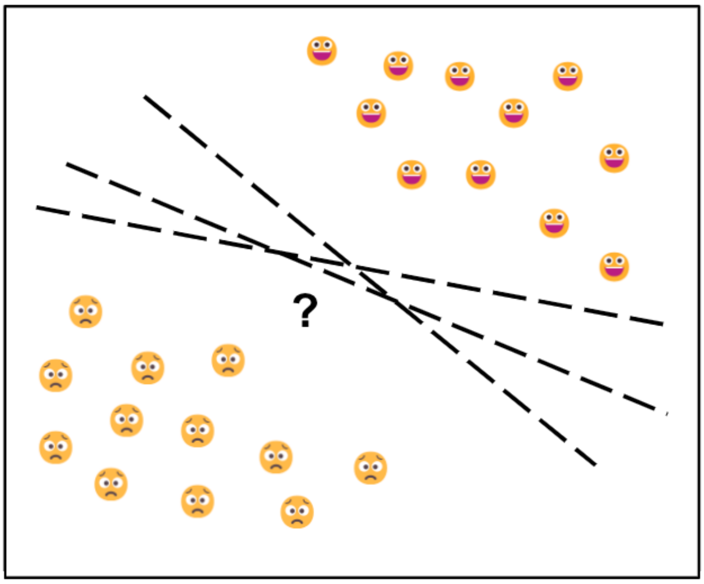

Figure 1: Two-class linear separation problem: how to choose the optimal hyperplane from all those that can separate the data?

Hyper-parameters are configurations external to the model, often defined prior to training. They are not learned directly from the data, but are adjusted manually through testing or optimization techniques. Support vector machines have several hyper-parameters.
Support vector machines have several hyper-parameters: C, Kernel, Gamma, degree, coefficient... Using an SVM with an RBF kernel often yields excellent results for classification tasks across various datasets. However, it's crucial to fine-tune the hyper-parameters C and γ according to the specific dataset [17]. Where C (type: Scalar) is the regularization parameter in SVM , it balances between maximizing margin and minimizing the training error, while γ (type: Scalar) decides that how much curvature we want in a decision boundary. Hyper-parameter selection poses a non-convex optimization challenge, prompting the development of numerous algorithms to address it. These include grid search, random search, Bayesian optimization, simulated annealing, particle swarm optimization, Nelder-Mead, and various others [18].

3.2. Termite Alate Optimization Algorithm (TAOA)

The study in [5] proposed an optimization algorithm named Termite Alate Optimization Algorithm (TAOA), based on the behavior of phototactic of the termite alate group. The proposed algorithm follows two main rules: (1) Alates are drawn towards areas with the highest brightness while being deterred by those with the lowest brightness (2) The quantity of Alates seeking the brightest location stays consistent. Alates situated in darker regions face risks such as predation by birds or wing loss, prompting their replacement with new alates, the Figure 2 illustrates this behaviour. Each alate possesses a brightness, which is determined by its fitness function. In the first phase, each alate seeks to move towards the alate positioned in the best position while simultaneously moving away from the alate in the darkest place. The equation used to update the position of the i^{th} alate at the t^{th} Equation 2.

$$\begin{aligned} &alate_i^{t+1} = alate_i^t + r1(alate_{best}^t - alate_i^t) - r2(alate_{worst}^t - alate_i^t) \\ &\text{if } alate_i^t \neq alate_{best}^t \end{aligned} \tag{2}$$

Where r1 and r2 have random values between 0 and 1.

In the second phase, a specified percentage (Pe) of alates located in regions of lower luminosity is removed from the population. If the selected alate occupies a brighter position, it replaces the existing one. The 3 is used to select the new alate.

$$alate_{new} = alate_a + \beta(alate_b - alate_c) \quad (3)$$

Where $\mathrm{alate_a}$, $\mathrm{alate_b}$ and $\mathrm{alate_c}$ are random alates from the best alates group. β is the adaptation factor with a value between 0 and 1. A lower value of β improves the exploitation process by favoring the selection of $\mathrm{alate_{new}}$ closer to the best alates.

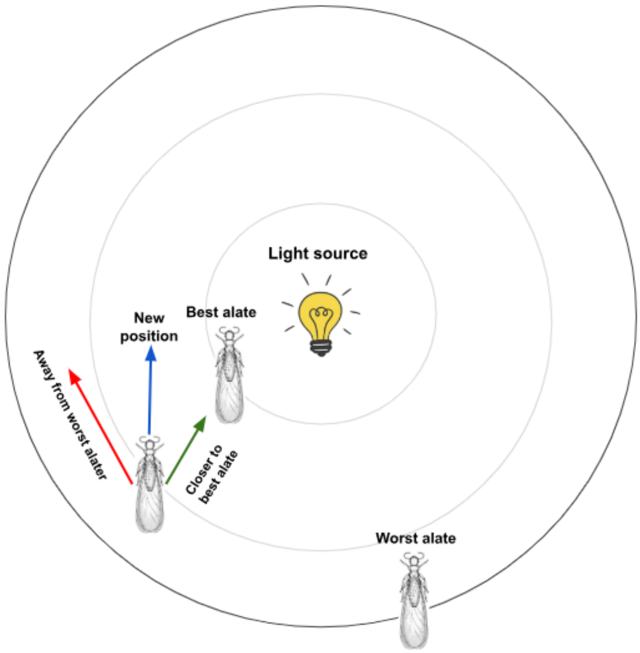

Figure 2: Alate motion in TAOA

4. TAOA for SVM hyper-parameter optimization

In this study, we aim to utilize TAOA to optimize the two hyper-parameters C and gamma in SVM for sentiment analysis. The diagram in Figure 3 illustrates the sequential flow from pre-processing to the optimization phase, showcasing our methodology optimizing the SVM's hyper-parameters in the goal of enhancing SVM model performance.

4.1. Preprocessing phase

The pre-processing phase is composed of the following tasks:

- Conversion to lowercase using Python's *lower()* function.
- Removal of HTTP links using the *split()* function to filter out words containing specific prefixes.
- Cleaning up special characters, numbers, and normalizing spaces using regular expressions from Python's "re" module.
- Tokenization using NLTK's *word_tokenize* function to segment text into tokens.
- Removal of stop words using NLTK's stopwords module.
- Stemming each token to its linguistic root using NLTK's PorterStemmer class.

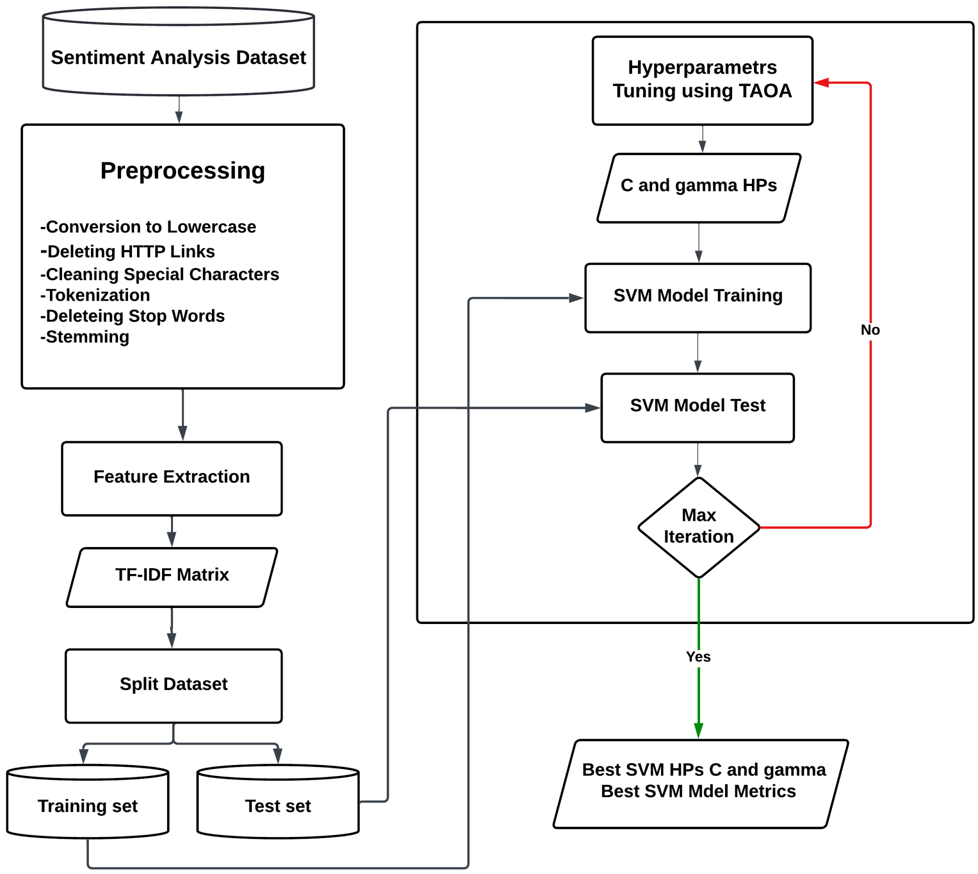

Figure 3: Methodology for SVM hyper-parameter tuning using TAOA

4.2. Hyper-parameter optimization and training phase

After the preprocessing phase, the TF-IDF dataset is partitioned into training dataset (80%) and testing dataset (20%). The TAOA optimization algorithm is called to search the optimal values for the two hyper-parameters C and gamma in several iterations. The best values obtained by TAOA are used by SVM for the training using the training TF-IDF dataset for generating a classification model. The performance of the model is then evaluated by using the test dataset.

4.2.1. TAOA for optimizing SVM's hyper-parameters

For resolving the HPs optimization problem by TAOA algorithm [5], a swarm of N Alates (particles) is considered, each particle representing a potential solution in 2-dimensional space for the two HPs C and gamma and randomly initialized in the search space, and the the TAOA algorithm is processed in T iterations. N and T are the two main parameters of TAOA. The main steps of TAOA for HPs optimization is presented in Algorithm 1.

TAOA's particle position encoding and initialization

Because the HPs C and gamma are continuous parameters in rages **[0.1, 1000]** and **[0.001, 100]** respectively, the position of each particle in TAOA is encoded as a 2-dimensional real vector, the first one for the C HP and the second for the gamma HP. The initial values of each of N particles of TAOA

Algorithm 1 Pseudo Code Termite Alate

$begin$
$Generate\ Alate^t \in \{alate_1^t, alate_2^t, ..., alate_n^t\}$
for $t = 1\ to\ t_{max}$ **do**
    while $i <= N$ **do**
        $Copmute\ fitness\ value\ (f_i^t)\ of\ alate_i^t$
        $i = i + 1$
    end while
    $Find\ alate_{best}^t\ and\ alate_{worst}^t$
    while $i <= N$ **do**
        if $alate_i^t \neq alate_{best}^t$ **then** $alate_i^{t+1} = alate_i^t + r1(alate_{best}^t - alate_i^t) - r2(alate_{worst}^t - alate_i^t)$
            $Copmute\ fitness\ value\ (f_i^t)\ of\ alate_i^t$
        end if
        $i = i + 1$
    end while
    $Sort \left\{alate_1^{t+1}, alate_2^{t+1}, ..., alate_n^{t+1}\right\}\ from\ best\ to\ worst$
    $G_s = P_e * N$
    $SP = N - G_s$
    $Alate_{best} \leftarrow \left\{alate_1^{t+1}, alate_2^{t+1}, ..., alate_{SP}^{t+1}\right\}$
    $Alate_{worst} \leftarrow \left\{alate_{SP+1}^{t+1}, alate_{SP+2}^{t+1}, ..., alate_N^{t+1}\right\}$
    while $j <= G_s$ **do**
        $Select\ alate_a, alate_b, alate_c\ Rndomly\ from\ Alate_{best}$
        $alate_{new} = alate_a + (alate_b - alate_c)$
        $Copmute\ fitness\ value\ (f_{new})\ of\ alate_{new}$
        if $f_{new}\ is\ better$ **then**f_{SP+j}^{t+1}
            $Replace\ alate_{SP+j}^{t+1}\ by\ alate_{new}\ in\ alate_{worst}$
        end if
        $j = j + 1$
    end while
    $alate^t \in alate_{best}, alate_{worst}$
end for
$Optimum\ solution = alate_{best}^t$

are randomly initialized i.e. set random values for C and gamma in their ranges.

Fitness function

The fitness function employed in the TAOA's algorithm aims to maximize the classification accuracy obtained by SVM on the TF-IDF test dataset. The fitness function is presented in Equation 4 . The SVM's accuracy is a crucial metric in determining the quality of the hyperparameter configuration. By maximizing the classification accuracy, the fitness function guides the swarm towards hyperparameter configurations that lead to improved model performance.

$$\text{Fitness }(\text{alate}_i) = SVMAccuracy \tag{4}$$

5. Experiments

This section summarizes the obtained result of the comparison of TAOA against default hyper-parameters and PSO.

5.1. Datasets

There are several datasets in the SA field, to evaluate the proposed approach; five datasets were selected due to their common use in the literature. Semeval_2016 (SE-2016) with 3198 tweets, Semeval_2017 (SE-2017) with (1252), Stanford with 2999 tweets, Polarity Movie Data (PMD) with 1001 tweets, and Movie Review Data (MRD) with 1200 tweets.

5.2. Evaluation metrics

We used the following five metrics in the experiment: Accuracy, precision, recall, F1-score and Computation time.

- **Accuracy**
  It is a metric that measures how often a machine learning model is correct overall. The accuracy is calculated as in Equation 5.

$$Accuracy(Acc) = \frac{\text{Correct Predictions}}{\text{All Predictions}} \tag{5}$$

- **Precision**
  It is a metric that measures how often a machine learning model correctly predicts the positive class. The precision value is calculated as in Equation 6.

$$Precision(Pre) = \frac{\text{True Positives}}{\text{True Positives + False Positives}} \tag{6}$$

- **Recall**
  It measures how often a machine learning model accurately identifies positive instances (true positives) among all actual positive samples present in the dataset. The recall value is calculated as in Equation 7.

$$Recall(Rec) = \frac{\text{True Positives}}{\text{True Positives + False negatives}} \tag{7}$$

- **F1-score**
  In scenarios where both precision and recall are equally critical, the F1-score serves as a valuable metric. The F1-score, calculated as the harmonic mean of precision and recall, provides a single measure that balances both precision and recall. A high F1-score means good precision value and good recall value . The F-score value is calculated as in Equation

$$F1 - score(F1) = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}} \tag{8}$$

- **Computation Time**
  In addition to the usual evaluation metrics: Accuracy, Precision, Recall and F1-score this section also considers the computational time of the algorithms. It is a crucial factor in our study.

Where, true positive are instances correctly identified as positive. False positives are the instances wrongly identified as positive. False negatives are the instances wrongly identified as negative.

5.3. Experimental Setup

In our analysis, we compared the performance of the TAOA algorithm with the PSO algorithm for optimizing the hyper-parameters of the SVM on five well-known datasets in sentiment analysis. For both algorithms, we utilized a configuration with a fixed number of particles set to 100 and a total of 20 iterations. Specifically, for PSO, we selected parameters C1 = 2, C2 = 2, and weight (w) = 0.3. Conversely, for the TAOA, we opted for parameters Pe = 0.25 and β = 0.7.

5.4. Results

The five tested metrics accuracy, precision, recall, F1-measure and Computation time of the SVM's model was evaluated using: first, the default values of the SVM's HPs that specified by the Python scikit-learn library package, second the HPs generated by the PSO algorithm [19], and finally the HPs generated by the TAOA optimization algorithm[5]. Table 1 presents the obtained results for each metric; while Table 2 shows the best values of the two SVM's HPs (C and gamma) obtained by each optimization algorithm in addition to the time conception of the two algorithms PSO and TAOA for optimization. From these results, it is clear that the TAOA algorithm enhanced the accuracy on all datasets SE-2016, SE-2017, PMD, and MRD by 4%, 2%, 5%, and 4%, respectively. The PMD dataset saw the highest increase in accuracy, rising from 0.77 to 0.82, while the Stanford dataset showed the smallest improvement, increasing from 0.69 to 0.6983. For the F1-Score metric as a result of improvements in both precision and recall, The TAOA algorithm improved it in all datasets except SE-2016, it means that the SVM model now performs better in correctly identifying positive tweets(precision) and becoming better at identifying more positive tweets out of all the tweets that are actually positive(recall). Additionally, it's noteworthy that the metrics achieved by TAOA were also attainable through the PSO algorithm. However, TAOA outperformed PSO in terms of Computation time, as it required less time to converge to optimal solutions. For instance, With SE-2016, the Computation time decreased to half, and with SE-2017, it was less than half compared to PSO.

Table 1
Comparison results

Dataset	**Default Values**				**PSO Metrcis**				**TAOA Metrics**			
	Accu	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
SE-2016	0.73	0.88	0.30	0.73	0.77	0.77	0.53	0.63	0.77	0.77	0.53	0.63
SE-2017	0.79	0.84	0.73	0.79	0.81	0.86	0.76	0.81	0.81	0.86	0.76	0.81
Stanford	0.69	0.64	0.76	0.70	0.69	0.65	0.78	0.71	0.69	0.65	0.78	0.71
PMD	0.77	0.76	0.78	0.77	0.82	0.80	0.84	0.82	0.82	0.80	0.84	0.82
MRD	**0.77**	0.84	0.72	0.78	0.81	0.85	0.79	0.82	0.81	0.85	0.79	0.82

Table 2
Hyper-parameter values and Computation Time

Dataset	Default Values		PSO HPs			TAOA HPs		
	C	gamma	C	gamma	Time(s)	C	gamma	Time(s)
SE-2016	1.0	1.001	659.42	0.001	2562	654.08	0.001	**1260**
S7-2017	1.0	1.002	458.87	1.3810	1574	980.06	0.7795	**977**
Stanford	1.0	1.0010	248.45	1.2325	6282	538.30	1.2260	**4561**
PMD	1.0	1.0072	747.24	0.0919	18065	92.10	0.0544	**11761**
MRD	1.0	1.0065	1000	0.001	30003	970.54	0.001	**18300**

6. Conclusion

This research aims to improve the performance of SVM models in sentiment analysis through hyper-parameter optimization. We focused on tuning the two main hyper-parameters of SVM (C and gamma) using Termite Alate Optimization Algorithm (TAOA). After experiments on five sentiment analysis datasets, we observed that the TAOA can increase the accuracy of SVM models in all datasets compared to SVM with default hyper-parameters on one hand, and exhibited quicker performance than PSO i.e. it completes its optimization process in less time. For future work, an improvement of TAOA by hybridization with other metaheuristics can significantly improve the performance.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] P. Nandwani, R. Verma, A review on sentiment analysis and emotion detection from text, Social network analysis and mining 11 (2021) 81.

[2] P. K. Jain, R. Pamula, G. Srivastava, A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews, Computer science review 41 (2021) 100413.

[3] D. A. Pisner, D. M. Schnyer, Support vector machine, in: Machine learning, Elsevier, 2020, pp. 101–121.

[4] K.-X. Han, W. Chien, C.-C. Chiu, Y.-T. Cheng, Application of support vector machine (svm) in the sentiment analysis of twitter dataset, Applied Sciences 10 (2020) 1125.

[5] A. Majumder, Termite alate optimization algorithm: a swarm-based nature inspired algorithm for optimization problems, Evolutionary Intelligence 16 (2023) 997–1017.

[6] B. Shekar, G. Dagnew, Grid search-based hyperparameter tuning and classification of microarray cancer data, in: 2019 second international conference on advanced computational and communication paradigms (ICACCP), IEEE, 2019, pp. 1–8.

[7] P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, J. R. Pastor, Particle swarm optimization for hyper-parameter selection in deep neural networks, in: Proceedings of the genetic and evolutionary computation conference, 2017, pp. 481–488.

[8] C. Witt, Worst-case and average-case approximations by simple randomized search heuristics, in: Annual Symposium on Theoretical Aspects of Computer Science, Springer, 2005, pp. 44–56.

[9] E. Elgeldawi, A. Sayed, A. Galal, A. Zaki, Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. informatics 2021, 8, 79, 2021.

[10] L. Villalobos-Arias, C. Quesada-López, J. Guevara-Coto, A. Martínez, M. Jenkins, Evaluating hyper-parameter tuning using random search in support vector machines for software effort estimation, in: Proceedings of the 16th ACM international conference on predictive models and data analytics in software engineering, 2020, pp. 31–40.

[11] I. Syarif, A. Prugel-Bennett, G. Wills, Svm parameter optimization using grid search and genetic algorithm to improve classification performance, TELKOMNIKA (Telecommunication Computing Electronics and Control) 14 (2016) 1502–1509.

[12] N. Rodríguez-Barroso, A. R. Moya, J. A. Fernández, E. Romero, E. Martínez-Cámara, F. Herrera, Deep learning hyper-parameter tuning for sentiment analysis in twitter based on evolutionary algorithms, in: 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), IEEE, 2019, pp. 255–264.

[13] A. D. N. W. Susanto, H. Suparwito, et al., Svm-pso algorithm for tweet sentiment analysis# besoksenin, Indonesian Journal of Information Systems 6 (2023) 36–47.

[14] N. K. Wardhani, S. K. Rezkiani, H. Setiawan, G. Gata, S. Tohari, W. Gata, M. Wahyudi, Sentiment analysis article news coordinator minister of maritime affairs using algorithm naive bayes and support vector machine with particle swarm optimization, J. Theor. Appl. Inf. Technol 96 (2018) 8365–8378.

[15] G. Deffuant, S. Martin, L. Chapel, Utiliser des" support vector machines" pour apprendre un noyau de viabilité, in: MajecSTIC 2005: Manifestation des Jeunes Chercheurs francophones dans les domaines des STIC, 2005, pp. 195–202.

[16] P. S. Hutapea, W. Maharani, Sentiment analysis on twitter social media towards shopee e-commerce through support vector machine (svm) method, JINAV: Journal of Information and Visualization 4 (2023) 7–17.

[17] R. Amami, D. B. Ayed, N. Ellouze, Practical selection of svm supervised parameters with different feature representations for vowel recognition, arXiv preprint arXiv:1507.06020 (2015).

[18] J. Wainer, P. Fonseca, How to tune the rbf svm hyperparameters? an empirical evaluation of 18 search algorithms, Artificial Intelligence Review 54 (2021) 4771–4797.

[19] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, volume 4, ieee, 1995, pp. 1942–1948.