A Comparative Study of Evolutionary Methods for Feature Selection in Sentiment Analysis

Shikhar Garg and Sukriti Verma
Adobe Systems, Noida, Uttar Pradesh, India

Keywords: Meta-heuristic, Feature Selection, Evolutionary Algorithm, Binary Bat, Binary Grey Wolf, Genetic Algorithm.

Abstract: With the recent surge of social media and other forums, availability of a large volume of data has rendered sentiment analysis an important area of research. Though current state-of-the-art systems have been demonstrated impressive performance, there is still no consensus on the optimum feature selection algorithm for the task of sentiment analysis. Feature selection is an indispensable part of the pipeline in natural language models as the data in this domain has extremely high dimensionality. In this work, we investigate the performance of two meta-heuristic feature selection algorithms namely Binary Bat and Binary Grey Wolf. We compare the results obtained to employing Genetic Algorithm for the same task. We report the results of our experiments on publicly available datasets drawn from two different domains, viz. tweets and movie reviews. We have used SVM, k-NN and Random Forest as the classification algorithms.

1 INTRODUCTION

Social networking platforms such as blogs, sites and other forums are being increasingly used by people to express sentiments, opinions and emotions in the form of reviews, comments, posts, tweets etc. However, because of the sheer volume of data that is available, understanding and distilling the information contained in this data in a meaningful manner remains a formidable task. Sentiment analysis, also known as opinion mining, is a field within natural language processing that concerns itself with this task of building systems to collect, examine and understand the sentiments and opinions expressed by people. It can be modelled as a form of classification of data with respect to sentiments: positive, negative or neutral. Sentiment analysis has various practical applications to businesses, marketers, advertisers etc. (Medhat et al., 2014)

There are several challenges in building sentiment analysis systems to analyze textual data in English language. Most of these issues stem from the peculiarities and ambiguities present in the language and the subtle differences in expressions that make it difficult to assess sentiments accurately (Vinodhini and Chandrasekaran, 2012). One major issue with text documents is the extremely high dimensionality of text data. Moreover, it is difficult to define the features of a text document as these are usually hidden in a large pool of subjective text (Eirinaki et al., 2012). Commonly used features are single word, character N-grams, word N-grams etc. In this work, we use TF-IDF feature representation and aim to leverage feature selection to improve and enhance sentiment analysis for text data in English (Forman, 2003).

Feature selection is a dimensionality reduction technique to remove irrelevant, redundant and noisy features from the dataset. It is an important step specifically for processing text data due to its dimensionality (Karabulut et al., 2012b). Feature selection often leads to a lower computation cost, improved learning accuracy and enhances model interpretability (Miao and Niu, 2016). The process of feature selection can be either independent or dependent on the classifier. The former type of methods are more time efficient in comparison but the latter type of methods are more reliable because while selecting features these methods consider the effect that a feature may have towards a specific classifier. Hence, this work focuses on methods of feature selection that are dependent on the classifier. However, the selection of an optimal subset of features with respect to classifier accuracy is an NP-hard problem (Novakovic, 2010). Some heuristic algorithms such as best first search, random search and evolutionary algorithms are often used (Ahmad et al., 2015). In this work, we employ two meta-heuristic algorithms, namely Binary Bat (Nakamura et al., 2012) and Binary Grey Wolf

Garg, S. and Verma, S.
A Comparative Study of Evolutionary Methods for Feature Selection in Sentiment Analysis.
DOI: 10.5220/0007948201310138
In *Proceedings of the 11th International Joint Conference on Computational Intelligence (IJCCI 2019)*, pages 131-138
ISBN: 978-989-758-384-1

(Emary et al., 2016), for the task of feature selection. Research has shown that these algorithms tend to outperform various conventional algorithms (Fong et al., 2013). We investigate these methods for feature selection and compare them to basic evolutionary methods. We carry out this study on two publicly available datasets.

The main contributions of this work over existing literature are as follows:

1. We compare the performance of meta-heuristic algorithms with basic evolutionary algorithms.
2. We also compare two closely linked meta-heuristic evolutionary algorithms hence providing some insight into the subtle differences that exist between the two.

2 RELATED WORK

Recent boom in the usage of social networking sites and the availability of a large amount of data has deemed sentiment analysis an important area of investigation. There are multiple granularities at which sentiment based classification of text may be performed: the document level, sentence level, or attribute level (Vinodhini and Chandrasekaran, 2012). In this work, we focus on document level sentiment classification of tweets drawn from the Senti140 dataset (Go et al., 2009) and movie reviews drawn from Cornell Movie Reviews dataset (Pang and Lee, 2004). The main approaches undertaken to perform sentiment analysis in exiting literature can be categorized into 2 categories:

1. **Lexicon based Approach:** This type of approach uses out-of-context scoring for individual features followed by combining these individual scores into a score for the entire document.
2. **Machine Learning based Approach:** This type of approach uses predictive models to learn the mapping from document feature values to the sentiment of the document.

Machine learning based approaches are much more accurate than lexicon based approaches as the latter do not include any contextual information while deciding the sentiment of a given document. Work by Pang et al. (Pang et al., 2002) first modelled sentiment classification as a special case of topic based document classification and applied supervised machine learning models to perform the same. They demonstrated Naive Bayes (Tan et al., 2009), Maximum Entropy, and Support Vector Machines (Mullen and Collier, 2004) to perform better than any other approach at the time. The main step before training any machine learning model is to engineer a set of features that are given as input. As discussed before, one of the challenging issues in text classification is presence of a large number of features. As each feature can affect the classification accuracy positively or negatively, feature selection methods are required to retain only the most relevant and informative features (Forman, 2003).

Feature selection methods may be divided into three categories (Ahmad et al., 2015):

1. **Filtration:** This type of feature selection is independent of the machine learning algorithm. It aims to assign an importance score to each feature using statistical analysis.
2. **Wrapping:** This type of approach is dependent on the machine learning algorithm. Due to this dependence, it requires extensive computation but is a more reliable method than filtration because it considers the effect that a feature may have on a given classifier.
3. **Hybrid:** This type of approach overcomes the weaknesses of both filters and wrappers by using a combination of the two to limit computation while interacting with the classifier.

Conventional feature selection methods have been widely studied for some time in the research community and their effects on various types of datasets have also been well explored (Karabulut et al., 2012a). Most conventional feature selection methods were of the filtration category. Principal Component Analysis is one such feature selection algorithm which exploits the concept of co-variance matrix and eigen-vectors to determine the most important features (Song et al., 2010). Study has shown that when this algorithm is applied on face recognition task it reduces the dimensionality significantly and accuracy is also not adversely affected (Song et al., 2010). Another conventional feature selection algorithm, Chi-square, has been found to be successful on the task of Arabic text categorization and yielding an F measure of 88.11 (Mesleh, 2007). Other methods that have been successfully applied to sentiment analysis for English text are Document frequency, Information Gain, Gain Ratio, Correlation-based Feature Selection, Mutual Information, Fisher Score, Relief-F etc. (Sharma and Dey, 2012) (Forman, 2003). These methods do not interact with the classifier.

However, in the last decade, the trend has shifted to wrapper category of methods that have interaction with the classifier. To reduce computation within the wrapper based feature selection methods, meta-heuristic search can replace brute-force search. Meta-

heuristic algorithms have their basis in the behavior of biological systems present in nature and are capable of searching large state spaces. These algorithms have been successfully applied to problems that require combinatorial exploration (Ahmad et al., 2015). Evolutionary meta-heuristic algorithms like genetic algorithm have shown reasonably positive results when employed to the task of high-dimensionality feature selection by searching for optimal feature subset stochastically (Zhu et al., 2010). The two main components of these algorithms are diversification and intensification (Gandomi et al., 2013). While intensification causes successful exploitation of the locally best solutions, diversification ensures that the algorithm does an efficient global search and explores the entire state space to a sufficient measure. These two processes ensure that the search space is searched thoroughly. The primary objectives behind the development of these algorithms are efficiently solving search problems with a large state space and obtaining methods that are more robust than the current ones (Talbi, 2009).

Feature selecting using Genetic Algorithm in combination with Information Gain (Abbasi et al., 2008), has been demonstrated to achieve very high accuracies of over 93% for multiple languages. Ant Colony optimisation, a nature inspired meta-heuristic swarm intelligence algorithm (Fong et al., 2013), has been demonstrated to perform well for high-dimensionality feature selection. It has been shown to outperform Genetic Algorithm, Information Gain and Chi-square method on the task of feature selection on the Reuters-21578 dataset (Aghdam et al., 2009). Other more sophisticated meta-heuristic algorithms yield an improved accuracy even in combination with very simple classifiers such as optimum path forest (Rodrigues et al., 2014).

In this publication, we investigate the performance of two meta-heuristic algorithms: Binary Bat (Nakamura et al., 2012) and Binary Grey Wolf (Emary et al., 2016) for the task of feature selection. The classification accuracy of these two feature selection methods is explored using three machine learning classifiers: Random Forest, Support Vector Machine and k-Nearest Neighbor. The remainder of the paper is organized as follows. Section 3 outlines our approach in detail. Section 4 reports performance of these two meta-heuristic feature selection algorithms when used with 3 classifiers on two publicly available datasets. Section 4 also compares the proposed technique with basic evolutionary techniques for feature selection. Sections 5 concludes this publication.

3 APPROACH

In this section, we provide a detailed description of our proposed approach. It can be broken down into 3 major phases:

1. **Pre-processing and Feature Extraction:** This phase comprises of the steps required to extract relevant features from a given text document.
2. **Feature Selection:** This phase includes application of meta-heuristic algorithms to reduce dimensionality of the feature space.
3. **Classification:** This is where we train a machine learning classifier to predict the sentiment of a given text document.

3.1 Preprocessing and Feature Extraction

Preprocessing is crucial when it comes to processing text. In this phase, we do the following:

1. **Paragraph Segmentation:** The paragraphs in a document are broken down into sentences.
2. **Word Normalization:** Each sentence is broken down into words and the words are normalized. Normalization involves lemmatization and results in all words being in one common verb form, crudely stemmed down to their roots with all ambiguities removed. For this purpose, we use Porters algorithm.
3. **Stop Word Filtering:** Each token is analyzed to remove high frequency stop words.

The next step is to map given input to some set of features that distill crucial information present in the given document. There is still no consensus on what type of features would serve as the best representation of a text document for the task of classification. The feature representations proposed in the literature belong to 3 types: semantic, stylistic, and syntactic (Abbasi et al., 2008). Different studies have tried to resolve this issue and compared existing feature selection methods (Pang et al., 2002). Most of the existing research uses simple features like single words, character N-grams, word N-grams or some combination of these (Abbasi et al., 2008). In this work, we use TF-IDF feature representation (Salton and Buckley, 1988). The use of TF-IDF to represent textual features has been thoroughly justified in literature (Hiemstra, 2000).

3.2 Feature Selection

In this section, we briefly outline the two algorithms that have been adapted for feature selection in this work.

3.2.1 Binary Bat Algorithm

The advanced capability of echolocation in bats has been used to develop a meta-heuristic optimization technique called the Bat Algorithm (Yang, 2012). Echolocation works as a type of sonar: bats, emit a loud and short pulse of sound, wait till the sound hits an object and the echo returns back to their ears (Griffin et al., 1960). The time it takes for the echo to reach back lets bats compute how far they are from the said object (Metzner, 1991). Furthermore, bats have a mechanism to distinguish the difference between an obstacle and a prey (Schnitzler and Kalko, 2001).

The Bat Algorithm encodes the behavior of a band of bats tracking prey using echolocation. In order to model this algorithm, (Yang, 2012) has idealized some rules, as follows:

1. All bats use echolocation to sense distance, and they can also differentiate between food, prey and obstacles.
2. A bat b_i flies with velocity v_i at position x_i with a fixed frequency F_{min}, varying wavelength λ and loudness A_0 to search for prey. They automatically adjust the wavelength of their emitted pulses. They also adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity to their target.
3. Although the loudness can vary in many ways, Yang (Yang, 2012) assumes that the loudness varies from a large positive value, A_0 to a minimum constant value, A_{min}.

Each bat b_i is assigned a random initial position x_i, velocity v_i and frequency f_i. At each time step t, updates to the state of each bat are made using the following equations:

$$f_i = f_{min} + (f_{min} - f_{max})\beta \tag{1}$$

$$\vec{V}_i(t) = \vec{V}_i(t-1) + [\hat{X} - \vec{X}_i(t-1)]f_i \tag{2}$$

$$\vec{X}_i(t) = \vec{X}_i(t-1) + \vec{V}_i(t) \tag{3}$$

Here, β denotes a randomly generated number drawn from [0,1]. Frequency f_i is used to control the pace of the movement. $\hat{X}$ denotes the current global best solution. After each iteration, the loudness A_i and the pulse rate r_i are updated as follows:

$$A_i(t+1) = \alpha A_i(t) \tag{4}$$

Table 1: Hyperparameters for the Binary Bat Algorithm.

Hyperparameter	Value
Number of Bats	70
Number of Iterations	100
Alpha, α	0.90
Gamma, γ	0.90
F_{min}	0.00
F_{max}	1.00
$A_i(0)$	1.5
$r_i(0)$	0.5

$$r_i(t+1) = r_i(0)[1 - exp(-\gamma t)] \tag{5}$$

Here, α and γ are hyperparameters. The suggested values for $A_i(0) \in [1,2]$ and $r_i(0) \in [0,1]$ (Yang, 2012). Yang (Yang, 2012) also suggests the concept of random walks to introduce and improve divesity in the current solution set. With feature selection, we are working in a discrete space where we want to either keep a feature or discard it. To this end, Nakamura et al. (Nakamura et al., 2012) have developed a binary version of the Bat Algorithm. They apply the sigmoid function to limit a bat's position to binary values. The hyperparameters used in this work are listed in Table 1. These hyperparameters were tuned experimentally.

3.2.2 Binary Grey Wolf Optimisation Algorithm

Grey Wolfs have a very interesting social behaviour. They often live in a pack and follow a very rigid social hierarchy of dominance. At the top of hierarchy are their leaders which are known as the alphas. They dictate orders which must be followed by the group. Just below the alphas are the betas. They are strong wolfs which ensures that alpha's order is followed and are in next line to become the alpha. The subset of wolves which are dominated by all the other wolves are known as omegas. The remaining wolves which are not alpha, beta or omega are deltas.

The social behaviour of interest to this algorithm is the hunting behaviour of the wolves. The key stages of hunting are as follows (Muro et al., 2011):

1. Approaching the prey
2. Encircling the prey
3. Attacking the prey

Mathematical modelling of social hierarchy sets the fittest candidate solution as the alpha α, and subsequently the second fittest solution as beta β, the third fittest candidate solution as delta δ. The rest of the solutions in the search space are set as omegas ω. Hunting is guided by the positions of α, β, δ.

1. Approaching the prey is mathematically modelled by the change in the position vector of the grey

wolf. This position vector is denoted by $\vec{X}(t)$, where t is the iteration number.

2. Encircling of the prey is modelled using the following equations (Mirjalili et al., 2014):

$$\vec{D} = \left|\vec{C}.\vec{X}_p(t) - \vec{X}(t)\right| \quad (6)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \quad (7)$$

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \quad (8)$$

$$\vec{C} = 2\vec{r}_2 \quad (9)$$

where $\vec{D}$ denotes the distance between the wolf and the prey, $\vec{A}$ and $\vec{C}$ are coefficients vectors that control the exploitation and exploration phases of the grey wold optimization algorithm by using random vectors $\vec{r}_1$ and $\vec{r}_2$ and $\vec{a}$, which linearly decreases form 2 to 0 over the course of iterations.

3. Hunting behaviour is modelled by the following equations (Mirjalili et al., 2014):-

$$\vec{D}_\alpha = \left|\vec{C}_1.\vec{X}_\alpha(t) - \vec{X}(t)\right| \quad (10)$$

$$\vec{D}_\beta = \left|\vec{C}_2.\vec{X}_\beta(t) - \vec{X}(t)\right| \quad (11)$$

$$\vec{D}_\delta = \left|\vec{C}_3.\vec{X}_\delta(t) - \vec{X}(t)\right| \quad (12)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.\vec{D}_\alpha \quad (13)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2.\vec{D}_\beta \quad (14)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3.\vec{D}_\delta \quad (15)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (16)$$

The search process is initiated with the creation of a random population. At every iteration, α, β, and δ are responsible to estimate the likely position of the prey. The position of every candidate solution is updated according to the estimated position of the prey using equation 16. As equation 16 is dependent on the positions of the α, β, and δ, this algorithm mathematically models the fact that hunting is guided by these three wolves.

As discussed above, the problem of feature selection is modelled in a discrete space to either select a given feature or discard it, and hence (Emary et al., 2016) have come up with a binary version of this algorithm constraining the position of the wolves only to binary values.

$$X_i^{t+1} = Crossover(x_1, x_2, x_3) \quad (17)$$

Equation 17 represents the crossover between x, y, z and x_1, x_2, x_3. x_1, x_2 and x_3 are binary vectors depicting the move of ω towards α, β, δ. Calculations of x_1, x_2 and x_3 are shown below in equations 18, 21 and 24.

Table 2: Hyperparameters for the Binary Grey Wolf Optimisation Algorithm.

Hyperparameter	Value
Number of Wolves	70
Number of Iterations	100

$$x_1^d = \begin{cases} 1 & (x_\alpha^d + bstep_\alpha^d) \geq 1 \\ 0 & otherwise \end{cases} \quad (18)$$

$$bstep_\alpha^d = \begin{cases} 1 & cstep_\alpha^d \geq rand \\ 0 & otherwise \end{cases} \quad (19)$$

$$cstep_\alpha^d = \frac{1}{1 + e^{-10(A_1^d D_\alpha^d - 0.5)}} \quad (20)$$

$$x_2^d = \begin{cases} 1 & (x_\beta^d + bstep_\beta^d) \geq 1 \\ 0 & otherwise \end{cases} \quad (21)$$

$$bstep_\beta^d = \begin{cases} 1 & cstep_\beta^d \geq rand \\ 0 & otherwise \end{cases} \quad (22)$$

$$cstep_\beta^d = \frac{1}{1 + e^{-10(A_1^d D_\beta^d - 0.5)}} \quad (23)$$

$$x_3^d = \begin{cases} 1 & (x_\delta^d + bstep_\delta^d) \geq 1 \\ 0 & otherwise \end{cases} \quad (24)$$

$$bstep_\delta^d = \begin{cases} 1 & cstep_\delta^d \geq rand \\ 0 & otherwise \end{cases} \quad (25)$$

$$cstep_\delta^d = \frac{1}{1 + e^{-10(A_1^d D_\delta^d - 0.5)}} \quad (26)$$

where x_α^d, x_β^d and x_δ^d represents the position vector of the α, β and δ wolf respectively. $bstep_\alpha^d$, $bstep_\beta^d$ and $bstep_\delta^d$ are the binary step of the α, β and δ wolf respectively in dimension d. $cstep_\alpha^d$, $cstep_\beta^d$ and $cstep_\delta^d$ are the continuous step of the α, β and δ wolf respectively in dimension d and calculated using the sigmoid function. Finally, *rand* is a random number drawn from uniform distribution ∈ [0,1]. The hyperparameters used in this work are listed in Table 2.

3.3 Classification

The final component of our approach is the classifier. The classifier interacts with the feature selection algorithm, guiding it towards maximal accuracy. The testing accuracy of the classifier is used as the fitness function of an entity within our wrapper based feature selection approach. The performance of this approach is investigated using three machine learning classifiers: Random Forest, Support Vector Machine and k-Nearest Neighbor. In the next section, we document the results.

Table 3: Hyperparameters for the Genetic Algorithm.

Hyperparameter	Value
Number of Individuals	70
Number of Generations	100
Mutation Probability	0.25
Crossover Probability	0.5

4 RESULTS AND EXPERIMENTS

4.1 Experimental Setup

These comparative studies are done on the **Senti140** (Go et al., 2009) and **Cornell Movie Reviews** (Pang and Lee, 2004) datasets. The datasets were partitioned into three sets: a training set, a scoring set and a held-out validation set. The scoring accuracies were used to guide the fitness function of the feature selection algorithms. The final classification accuracy was reported on a held-out validation set that the classifier has not seen during the phase of feature selection or training.

1. **Senti140 Dataset:** This dataset contains 160,000 tweets that have been pre-processed to remove emoticons. The dataset has two target labels: 0 for negative and 4 for positive. We have bootstrapped and used 40k tweets divided equally among positive and negative target labels. After preprocessing, the number of features found in this dataset were 474056.
2. **Cornell Movie Reviews:** This dataset contains 5331 positive and negative movie reviews. In this study, we have used 5000 positive reviews and 5000 negative reviews for the task of training and testing the approach. After preprocessing, the number of features found in this dataset were 188440.

For each of the 3 classifiers, we run experiments with the 2 feature selection techniques discussed above. We compare the performance of using these 2 meta-heuristic algorithms for feature selection to using Genetic Algorithm for the same task (Ghosh et al., 2010). We employ one-point crossover for mating individuals and the old generation is completely replaced by the new generation. The individuals chosen for mating are selected by using a k-way tournament using k = 3. Other hyperparameters used for the Genetic Algorithm are listed in Table 3. For the results to be comparable and take up similar running time, the number of entities in one iteration/generation and the number of iterations/generations have been kept the same for all three feature selection methods.

Table 4: Values with SVM as classifier.

Dataset	Approach	Accuracy	Feature Reduction
Senti140	No FS	76.71%	-
	GA	80.11%	50%
	Binary Bat	77.35%	39%
	Binary Wolf	78.52%	17%
Cornell Movie Reviews	No FS	75.02%	-
	GA	80.92%	49.8%
	Binary Bat	77.45%	43%
	Binary Wolf	77.46%	19.5%

Table 5: Values with RF as classifier.

Dataset	Approach	Accuracy	Feature Reduction
Senti140	No FS	72.29%	-
	GA	73.81%	50.14%
	Binary Bat	72.97%	42.8%
	Binary Wolf	72.95%	22.5%
Cornell Movie Reviews	No FS	69.92%	-
	GA	70.72%	49.9%
	Binary Bat	70.28%	33.2%
	Binary Wolf	70.17%	15.3%

4.2 Results

In this section, we report the performance of Binary Bat, Binary Grey Wolf and Genetic Algorithm for the task of feature selection. We also report classifier accuracy without any feature selection method as a baseline. Table 4 reports the values when using SVM as the classifier. Table 5 reports the same for Random Forest and Table 6 reports the values for k-NN.

We can note all three methods of feature selection to consistently lead to an increased accuracy over the baseline when using SVM as the classifier. While the performance of the three methods among themselves are more or less, at par, Genetic Algorithm obtained the highest accuracies with gain as much as 5%. The difference in the number of features discarded is significant, with Genetic Algorithm achieving well over 2 times the reduction achieved by Binary Grey Wolf Algorithm. While Binary Bat Algorithm has behavior similar to swarm optimisation, Binary Grey Wolf Algorithm is guided by the 3 leading wolves and does not have an explicit diversification/intensification process going on. Hence, it is somewhat more vulnerable to get stuck in a local minima than the other methods and the average outcome may come to depend on the random initialisation.

When using RF as the classifier, the accuracy gain is not significant. This reasons conveniently from the fact that Information Gain is used as an implicit feature selector within each tree of the random forest

Table 6: Values with k-NN as classifier.

Dataset	Approach	Accuracy	Feature Reduction
Senti140	No FS	50.07%	-
	GA	63.00%	49.97%
	Binary Bat	59.69%	50.02%
	Binary Wolf	56.36	11.67%
Cornell Movie Reviews	No FS	58.28%	-
	GA	55.96 %	50.23%
	Binary Bat	55.20%	0.2%
	Binary Wolf	53.98%	8.4%

with max-depth being used to limit the number of features used. However, the computation of Information Gain that has to be done for every feature would significantly speed up with a lesser number of features. Genetic Algorithm again turns out to be the most efficient in terms of feature reduction.

When using k-NN as the classifier, the results seem to be mixed. While more than significant accuracy gains over the baseline have been obtained on the **Senti140** Dataset, we also observe worsened performance over the baseline for the **Cornell Movie Reviews** Dataset. This is probably because of the random nature of k-NN as it simply performs a majority voting within k-nearest neighbors and does not actually pick up any patterns. This shows that for some classifiers, any sort of feature selection will not guarantee an increase in the accuracy.

5 CONCLUSION

In this paper, we have compared the performances of meta-heuristic and evolutionary feature selection methods to the problem of sentiment analysis using various classifiers on two different domains of tweets and movie reviews. While we can see, that methods such as Random Forest that have in-built parameters to limit the features used, do not gain any sufficient improvement in accuracy, other methods such as SVM and k-NN can have gain in accuracy upto 25%. While the performance of Binary Bat and Genetic Algorithm was similar in terms of accuracy gain, the performance of Binary Grey Wolf Algorithm was consistently lower than these two. Also, the percentage decrease in the number of features is another important ground to consider while making a choice. Genetic Algorithm was observed to be the most efficient in terms of feature reduction percentage. Moreover, there is a difference in the number of hyperparameters that need to be tuned to make each algorithm work optimally, with Binary Grey Wolf Algorithm being the easiest to tune. Hence, a multitude of factors need to be considered when selecting a method for feature selection. The results reported in this paper can be used as a guidance for extended work in different domains.

REFERENCES

Abbasi, A., Chen, H., and Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):12.

Aghdam, M. H., Ghasem-Aghaee, N., and Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert systems with applications*, 36(3):6843–6853.

Ahmad, S. R., Bakar, A. A., and Yaakub, M. R. (2015). Metaheuristic algorithms for feature selection in sentiment analysis. In *2015 Science and Information Conference (SAI)*, pages 222–226. IEEE.

Eirinaki, M., Pisal, S., and Singh, J. (2012). Feature-based opinion mining and ranking. *Journal of Computer and System Sciences*, 78(4):1175–1184.

Emary, E., Zawbaa, H. M., and Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172:371–381.

Fong, S., Yang, X.-S., and Deb, S. (2013). Swarm search for feature selection in classification. In *2013 IEEE 16th International Conference on Computational Science and Engineering*, pages 902–909. IEEE.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305.

Gandomi, A. H., Yang, X.-S., and Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29(1):17–35.

Ghosh, S., Biswas, S., Sarkar, D., and Sarkar, P. P. (2010). Mining frequent itemsets using genetic algorithm. *arXiv preprint arXiv:1011.0328*.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12).

Griffin, D. R., Webster, F. A., and Michael, C. R. (1960). The echolocation of flying insects by bats. *Animal behaviour*, 8(3-4):141–154.

Hiemstra, D. (2000). A probabilistic justification for using tf$\times$ idf term weighting in information retrieval. *International Journal on Digital Libraries*, 3(2):131–139.

Karabulut, E., Özel, S., and Ibrikci, T. (2012a). Comparative study on the effect of feature selection on classification accuracy. *Procedia Technology*, 1:323 –327.

Karabulut, E. M., Özel, S. A., and Ibrikci, T. (2012b). A comparative study on the effect of feature selection on classification accuracy. *Procedia Technology*, 1:323–327.

Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

Mesleh, A. (2007). Chi square feature extraction based svms arabic language text categorization system. *Journal of Computer Science*, 3.

Metzner, W. (1991). Echolocation behaviour in bats. *Science Progress (1933-)*, pages 453–465.

Miao, J. and Niu, L. (2016). A survey on feature selection. *Procedia Computer Science*, 91:919–926.

Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69:46–61.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Muro, C., Escobedo, R., Spector, L., and Coppinger, R. (2011). Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations. *Behavioural processes*, 88(3):192–197.

Nakamura, R. Y., Pereira, L. A., Costa, K. A., Rodrigues, D., Papa, J. P., and Yang, X.-S. (2012). Bba: a binary bat algorithm for feature selection. In *2012 25th SIBGRAPI conference on graphics, patterns and images*, pages 291–297. IEEE.

Novakovic, J. (2010). The impact of feature selection on the accuracy of naïve bayes classifier. In *18th Telecommunications forum TELFOR*, volume 2, pages 1113–1116.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Rodrigues, D., Pereira, L. A., Nakamura, R. Y., Costa, K. A., Yang, X.-S., Souza, A. N., and Papa, J. P. (2014). A wrapper approach for feature selection based on bat algorithm and optimum-path forest. *Expert Systems with Applications*, 41(5):2250–2258.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Schnitzler, H.-U. and Kalko, E. K. (2001). Echolocation by insect-eating bats: We define four distinct functional groups of bats and find differences in signal structure that correlate with the typical echolocation tasks faced by each group. *Bioscience*, 51(7):557–569.

Sharma, A. and Dey, S. (2012). A comparative study of feature selection and machine learning techniques for sentiment analysis. In *Proceedings of the 2012 ACM research in applied computation symposium*, pages 1–7. ACM.

Song, F., Guo, Z., and Mei, D. (2010). Feature selection using principal component analysis. In *2010 international conference on system science, engineering design and manufacturing informatization*, volume 1, pages 27–30. IEEE.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.

Tan, S., Cheng, X., Wang, Y., and Xu, H. (2009). Adapting naive bayes to domain adaptation for sentiment analysis. In *European Conference on Information Retrieval*, pages 337–349. Springer.

Vinodhini, G. and Chandrasekaran, R. (2012). Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6):282–292.

Yang, X.-S. (2012). Bat algorithm for multi-objective optimisation. *arXiv preprint arXiv:1203.6571*.

Zhu, J., Wang, H., and Mao, J. (2010). Sentiment classification using genetic algorithm and conditional random fields. In *2010 2nd IEEE International Conference on Information Management and Engineering*, pages 193–196. IEEE.