Feature Selection based on Improved Differential Evolution (DE) Algorithm for E-mail Classification

Nadir Omer[1,*]

[1]Department of Information Systems and Cybersecurity, College of Computing and Information Technology, University of Bisha, Bisha 61922, Saudi Arabia

E-mail: nhamed@ub.edu.sa

Abstract

Spam e-mail has become a pervasive nuisance in today's digital world, posing significant challenges to efficient communication and information dissemination. Dealing with huge amounts of data with irrelevant and redundant features, which leads to high dimension. Nowadays, with the growth of using the internet, finding the secure E-mail classification system for cloud computing is a very important topic. Additionally, determining the best algorithm for choosing a subset of features has a big impact on how well automatic email classification works, making it one of the major issues. Among these is the Differential Evolution (DE) algorithm, which is computationally costly because of the slow convergence rate and evolutionary process. To address these issues, this study offers an intelligent scheme called Opposition Differential Evolution (ODE), which combines the Opposition Based Learning (OBL) and DE algorithms for effective automated feature subset selection. Its effectiveness is assessed using the support vector machine (SVM) to present a strong performance when evaluating the e-mail spam classification rate. Moreover, the OBL is used to accelerate and increase the convergence rate of traditional DE. To determine which features, contribute most to the reliability of the email spam classification, the subset features based on ODE that was used as feature subset selection are used.To assess the effectiveness of the proposed scheme, extensive experiments are conducted on spambase" and "spamassassin" benchmark email datasets, comprising a diverse collection of spam and non-spam emails. The results demonstrate that the Opposition Differential Evolution (ODE) algorithm yields superior performance compared to traditional machine learning and evolutionary techniques, displaying its robustness and efficiency in identifying spam emails accurately. The ODE algorithm effectively handles high-dimensional feature spaces, enhancing the model's discriminatory power while maintaining computational efficiency. Compared to the suggested ODE-SVM technique, which yields a result of 96.79 percent, the full-feature accuracy result was 93.55 percent. Additionally, empirical results demonstrate that our scheme may efficiently increase the number of features needed to improve the accuracy of the email spam classification.

Keywords: Feature Selection; E-mail spam classification; Opposition Based Learning (OBL), Differential Evolution (DE); Opposition Differential Evolution (ODE, Support Vector Machine (SVM)

1. Introduction

Due to the internet's explosive expansion and the pervasive adoption of email as a key communication tool, the issue of spam emails has been reached concerning proportions. In addition to inundating users' inboxes, unwanted and frequently harmful messages, or "spam," can present major security issues and impede effective communication. As such, efficient methods for classifying spam have become essential for guaranteeing the confidentiality and security of digital communications. Conventional approaches to spam classification, such as rule-based systems and keyword matching, have proven to be insufficient against the constantly changing strategies used by spammers. Because of this, scientists are using machine-learning algorithms to create spam classification systems that are increasingly intelligent and flexible. These methods offer a more thorough and automated method of classifying emails by attempting to detect spam using patterns and features extracted from

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

the email content. In the current digital era, spam emails have grown to be a persistent and harmful problem that compromises communication efficiency and poses major security risks [1]. To tackle this rising issue, strong and adaptable spam classification approaches are required due to the constant growth of spamming strategies. Conventional methods, including keyword matching and rule-based filters, have shown that they are inadequate in effectively detecting spam since they are unable to keep up with the constantly changing tactics used by spammers. By effectively separating spam from real emails using data-driven models, machine-learning algorithms have emerged as a potential approach to spam detection. Machine learning for spam classification has become a significant area of research in recent times [2]. Furthermore, machine learning techniques can extract knowledge from a collection of emails that are supplied [3]. By employing illustrative data, machine learning (ML) seeks to enhance computer program performance through experience, enabling smarter decision-making and intelligent problem-solving [4]. Then, one of the most crucial problems in the categorization of email spam is to solve the high dimensionality problem. In machine learning, two groups named feature extraction (feature transformation) and feature selection are used to extract features and prevent the high dimensionality of categorization [5]. The process of extracting new features from an original feature and changing arbitrary data into a unique feature space is known as feature extraction. Additionally, several techniques for extracting features and reducing their dimensionality. However, feature selection (FS) is a crucial method for reducing dimensionality and a crucial subject to choose a subset of features from the entire set of features [6]. Furthermore, not every element from an important level is equally important; some features may be redundant or useless [7]. Feature selection (FS) reduces the dimensionality of the feature space and removes redundant and irrelevant data to increase the effectiveness and efficiency of categorization [8]. FS differs from feature transformation in that it chooses a subset of the original attributes rather than creating new variables. Habitually, comparing the performance of classification algorithms before and after feature selection has been used to measure the effectiveness of feature selection algorithms. Based on the classification algorithm, evaluation metrics for feature selection strategies are separated into two groups: wrappers classifiers and filters. Since the filter method relies on some method of estimating the importance of individual features or subsets of features, it is typically much faster than the wrapper method. While wrapper depends on classification, algorithms and more accurately the using filter. However, the effectiveness of these algorithms is largely dependent on the feature extraction process and the optimization strategies applied to adjust the model's parameters. The capacity of nature-inspired optimization algorithms to effectively address complicated and high-dimensional optimization problems has led to their increasing popularity in recent years. The Opposition Differential Evolution (ODE) algorithm has become a highly effective optimization method among them. ODE is a novel approach that leverages a biological idea of opposition to improve the search performance of the traditional Differential Evolution (DE) method [9]. The fundamental purpose of this research is to present a unique way to spam identification by harnessing the capabilities of the Opposition Differential Evolution (ODE) algorithm. By leveraging the advantages of ODE in optimization, we aim to enhance the performance of email classification models and achieve improved accuracy and efficiency in distinguishing spam from legitimate emails. In this paper, we present a comprehensive study on the application of ODE in the context of spam detection. We begin by discussing the challenges associated with traditional spam filtering methods and highlight the need for more adaptive and intelligent solutions. Subsequently, we introduce the Opposition Differential Evolution (ODE) algorithm, explaining its underlying principles and advantages over traditional optimization algorithms. Furthermore, we outline the proposed methodology, which involves crucial steps such as feature extraction and selection to capture relevant characteristics from email content and the application of ODE to optimize the spam detection model's parameters. We also discuss the selection of appropriate benchmark datasets to accurately assess our method's performance [9]. We used a variety of datasets, including spam and non-spam emails, in our thorough trials to verify the efficacy of the suggested approach. We show how the ODE-based spam categorization system performs in comparison to traditional machine learning and optimization methods.

In conclusion, this paper contributes to the field of spam classification by leading an enhanced approach that utilizes the Opposition Differential Evolution (ODE) algorithm. We anticipate that our research will shed light on the benefits of nature-inspired optimization algorithms and inspire further investigations into the application of ODE in addressing other challenging problems in the domain of computational intelligence and machine learning. Ultimately, the successful application of ODE in spam detection could significantly improve email communication security and information integrity in the digital age. The remainder of the paper is structured as follows: In Section II, the contributions of ODE in the field of spam classification are highlighted together with a thorough assessment of comparable research in the field. The detailed technique for the Proposed Differential Evolution (DE) with opposition-based learning (OBL) is expounded upon in Section III. In Section IV, the experimental setup, results, and analysis are covered. Additionally, the benefits of employing ODE for spam classification are discussed. Section V provides a summary of our contributions and suggests future avenues for research, ending the paper. In conclusion, by presenting an improved method that makes use of the Opposition Differential Evolution (ODE) algorithm, this work seeks to advance the subject of spam classification.

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

We anticipate that our research will shed light on the benefits of nature-inspired optimization algorithms and inspire further investigations into the application of ODE in addressing other challenging problems in the domain of computational intelligence and machine learning. Ultimately, the successful application of ODE in spam detection could significantly improve email communication security and information integrity in the digital age.

2. Related Work

This section mainly discusses some of recently proposed e-mail spam classification techniques such as Different Evolution (DE) and opposition-based learning (OBL) [10]. These methods are under the category of evolutionary computation (EC) methods, which use SVM to increase classification accuracy and lower computational complexity in the process of classifying spam emails. For classification challenges, feature selection helps find the best subset of features from data samples. Furthermore, the improvement of classification accuracy by reducing the dimensionality of feature space using features selection methods or feature extraction methods to enhance the e-mail spam classification in terms of accuracy and computational complexity. As high-dimensional data continues to proliferate, Feature Selection (FS) has emerged as a crucial learning process activity. Important subjects in machine learning are feature ranking and feature selection. Getting rid of features that are redundant or superfluous might improve classification accuracy while accelerating processing. By identifying the subset of features that best classifies the training set, it streamlines the classification process and makes it easier to understand. This enhances the classifier's overall performance and helps it solve several issues. The goal of feature selection is to choose a representative subset of the original features while maintaining a high enough accuracy to reflect the original features, which contain a lot of noise, false information, and redundant and unnecessary characteristics. Then, one of the key methods for increasing classification accuracy is evolutionary computation (EC), which is divided into several families, including swarm intelligence (SI) and evolutionary algorithms (EA) groups [11]. Evolutionary algorithms (EAs) are widely recognized optimization techniques utilized to address complicated and complex issues [12]. Genetic algorithms (GA), differential evolution (DE), evolutionary programming (EP), genetic programming (GP), and evolutionary strategies (ES) are all examples of biological evolution and are included in the EA algorithm. Artificial intelligence techniques, or SIs, include particle swarm optimization (PSO), ant colony optimization (ACO), and bee colony algorithm (BCA). These techniques are based on the study of collective behavior in animal communities, such as those observed in fish, birds, and ants. In this study, we aim to enhance DE by using two new objective functions to achieve the best chromosome with highest fitness or lower objective function to select the optimal subset features. The best chromosome was achieved by improving the fitness function and using the output as input for classification based on SVM. Nevertheless, a few problems impede the effective use of emails. Email spam is one of them [2]. The more complex classifier-related problems have drawn the attention of numerous researchers studying the classification of email spam. In the classification of email spam, a variety of techniques and algorithms are employed to choose the significant subset features, including evolutionary computing algorithms and statistical techniques. An evolutionary computer technique called the evolutionary algorithm (EA) stores enough information about the population, search space, and features throughout the repeated search process.

Machine learning (ML) techniques can be embedded to examine this data to enhance the EAs search performance as compared to classical versions. ML extracts useful information to understand the search space behavior and further improve the upcoming search for its global optimum. In addition, research for the improvement of EAs through ML techniques was observed to have played an important role in literature EAs application adopting ML techniques have proven advantages in terms of solution quality and convergence speed. The literature is rich with different ML techniques that have been used for enhancing EA performance. In this paper opposition-based learning (OBL) techniques are used as it is widely applied for DE and has obtained high performances. The challenges that have arisen in enabling OBL to improve current e-mail spam classification problem are discussed later. In his review article [11] present more information on incorporating ML techniques for enhancing evolutionary computing (EC). The review shows that EC based application employing the ML approaches at each stage (population initialization, crossover, mutation, selection) enhances the quality solutions that are produced. Our literature review proposed that EAs in other domains were widely optimized using OBL to achieve better solutions than their contemporary traditional generation, particularly the DE algorithm. OBL is a learning technique that could prove that generating opposite solutions is optimum than random-based solutions. In other words, OBL is used to improve EAs which initialize their populations (solutions) using a random based approach for accessing the fittest solutions. Evolutionary algorithms (EAs) are one of evolutionary computing methods used to obtain the optimal solution. Additionally, EAs are computer programs designed to simulate biological evolution to solve complex problems. EA employs strategies including reproduction, mutation, recombination, and selection that are influenced by biological evolution. In the optimization problem, candidate solutions represent people in a population, and the environment in which the solutions "survivor" is determined by the fitness function (also see cost function). There are multiple components to evolutionary algorithms (EAs), including representation (individual definition), evaluation function (also known as fitness function), population, variation operators, parent

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

selection mechanism, recombination and mutation, and survivor selection mechanism. Differential evolution, genetic programming, and genetic algorithms are a few examples of evolutionary algorithms. An optimization and search technique called a genetic algorithm (GA) is founded on genetics and natural selection principles, making it a type of evolutionary algorithms (EA) method. DE is an effective algorithm that is straightforward and surpasses genetic algorithms (GAs) in solving numerous numerical single-objective optimization problems[13]. The primary distinction between DE and GA is that DE's convergence speed is faster than that of genetic algorithms [14]. Furthermore, one of the main problems with GA is that it still takes longer than DE. Additionally, while DE uses crossover to create a single trial vector, GA recombines two vectors to create two trail vectors. The selection process and the mutation system that makes DE self-adaptive are the primary distinctions between GA and DE. Regardless of their fitness value, all solutions in DE have an equal chance of being chosen as parents. The key distinction in building superior solutions is that DE relies on mutation operation, whereas genetic algorithms rely on crossover.

SVM is currently one of the most often used algorithms for classifying spam emails [15]. These days, handling infinite nonlinear features in high dimensional space is one of the key components of SVM. Then, because of the large amount of data, SVM frequently requires a lot of processing time and has a low classification accuracy rate [16, 17] . One of SVM's main issues is that it struggles to classify high-dimensional datasets with lots of features [8]. The result of the study by Priyanka et al. [2] on SVM for classifying large amounts of data shown that SVM is less accurate and requires more time when the amount of data is large. The evolutionary/stochastic character of population-based optimization methods often results in lengthy calculation durations, with DE being an exception [12]. The fundamental concept of OBL is to obtain better approximation for the current candidate solution by simultaneously considering an estimate and its corresponding opposite estimate (i.e., guess and opposite guess). Indeed, it has been demonstrated mathematically that opposing numbers are generally more likely to be near the ideal answer than merely random ones [12]. However, to explore the feature space, a search technique is required, such as particle swarm optimization or differential evolution (DE). Feature selection has received a lot of attention from data mining and machine learning researchers in recent years. The goal of feature selection in machine learning is to choose the best features with the highest accuracy [18]. To reduce dimensions and increase the classification rate of e-mail spam, this research suggested a hybrid ODE-SVM system that uses SVM as a classifier and opposition differential evolution (ODE) as feature selection. Opposition DE is employed as a search method because it quickly converges, yields the optimal (near-optimal) solution, and finds a useful subset of features to increase the SVM-based classification rate. The ODE-SVM strategy was created to increase the computational time complexity of SVM, decrease the large dimensionality of the feature space, and increase the accuracy of email spam classification. Additionally, this research is interesting since it is the first to use a mix of SVM as a classifier and ODE as feature selection for email databases. A comparison between the hybrid ODE-VM and SVM for email spam categorization has been conducted. According to experimental data, this approach has a low computing complexity and good classification accuracy. Figure 1 illustrates the suggested approach. This is how the rest of the paper is structured. The related work is discussed in Section 2. The suggested approach is explained in Section 3. Differential evolution (DE) is presented in Section 4. E-mail spam classification using ODE and SVM is presented in section five. Finally, the paper describes the experiment and discussions.

3. The proposed Differential Evolution (DE) with opposition-based learning (OBL)

Many complicated real-world problems have been successfully solved by improved systems in recent years. Since each intelligent system has its own weaknesses and an upgraded system is designed to compensate for them, the value of a unified system cannot be understated. The binary opposition differential evolution (ODE–SVM) scheme, which is the suggested approach, is divided into two sub-schemes: the SVM algorithm for classifying spam emails and the opposition differential evolution (ODE) scheme for feature selection. What is meant by "Binary" is the chromosomal arrangement that needs to be modified into binary dimension space. In the ODE–SVM schemes, every sub scheme functions as a distinct scheme and operates separately from the others. To choose the best (or almost best) subset of features, the ODE is trained. Then, using the SVM learning algorithm as classifiers for email spam, the outputs of the ODE are routed as input to the second scheme. The purpose of the ODE–SVM scheme is to evaluate the trained scheme's performance. We separated the dataset into 70% for training and 30% for testing because of the large number of samples connected to the datasets. The general phase of our suggested design is shown in Figure 1.

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

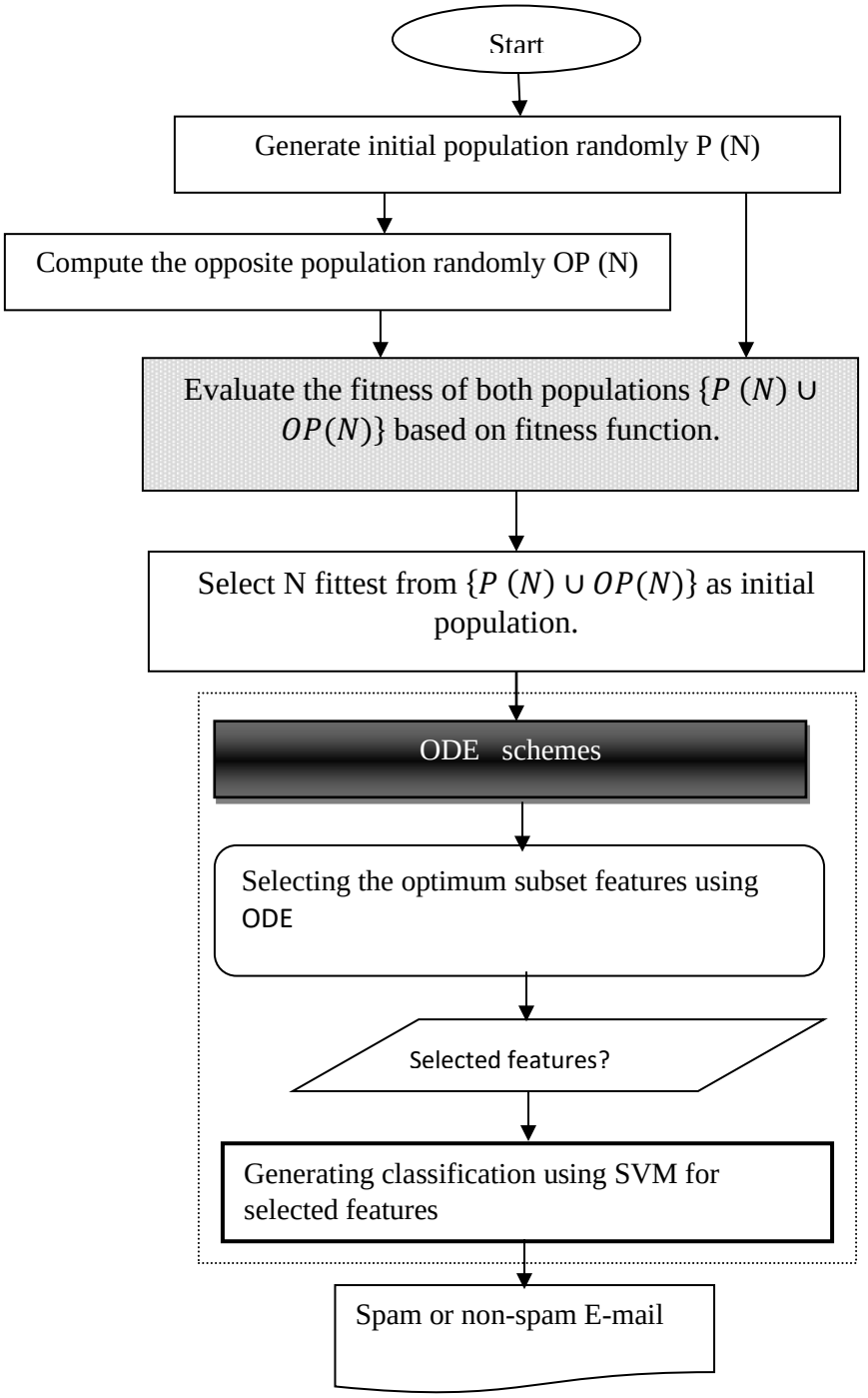

Figure 1. The overall procedure for the suggested proposed

Several researchers mention that the main issues in optimization techniques are random population generation and that the opposite values are sometimes too near to the optimal solution [12]. he majority of strategies used to enhance optimization techniques' search performance are machine learning techniques. In this study, the search space of DE has been improved by employing opposite numbers using OB. To choose the best subset characteristics and boost accuracy, the opposing DE was used as a feature selection tool. The EAs begin with a few beginning solutions (the initial population) and work their way up to one or more optimal answers. Numerous studies have been conducted in recent decades to enhance DE's performance. The OBL was used in this work to speed up the DE without causing stagnation or premature convergence [19]. ECs, like DE, are techniques that use concepts from biological evolution to produce initial solutions (population) at random. When scanning space, these algorithms typically overlook the prior knowledge about these solutions. Therefore, because the evolutionary

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

process is slow, EAs are computationally costly. Most optimization methods depend on population initialization; for optimal outcomes, intelligent initialization techniques based on machine learning techniques are needed to choose the appropriate solution. As far as we know, this is the first time opposing numbers have been used to enhance DE's feature selection for classifying spam emails. By utilizing random numbers and their opposites, OBL is used to speed up DE. OBL has been used to speed up DE's rate of convergence. Without introducing any more parameters, OBL is utilized to improve DE's performance. The impact of dimensionality has been examined using opposite numbers. Additionally, the convergence rate of the DE algorithms is easily affected, resulting in less precise control over their position and velocity. When starting a population and creating new populations during the evolutionary process, ODE employs opposite numbers. The learning and search process can be sped up by incorporating OBL into current soft computing methods. An efficient search for the intended solution resulted from optimal initialization. It's possible that algorithms won't find the target region or converge. It employs OBL in DE to increase fitness in terms of choosing the significant subset characteristics, accelerating convergence, and prevent parameter convergence. The modification of the current approach is the primary distinction between the improved DE in this study and the others. By employing opposing numbers, the pace of convergence of an optimization method has been enhanced. To generate an initial OBL-based population, the ODE follows the following steps see Figure 1:

1. Initialize the population P(N) randomly, N: is the population size.
2. Calculate opposites of DE $P_{opi}(N)$ using $P_{opi}(N) = a + b - P_i(N)$
3. Evaluate the fitness of both populations $\{P(N) \cup P_{op}(N)\}$ based on fitness function.
4. Choose N fittest individuals from set $\{P(N) \cup P_{op}(N)\}$ as initial population based on fitness value.
5. Iteration

a) Calculate mutation.
b) Calculate crossover.
c) Calculate the selection.

6. If the stopping criteria are met, output the best solution. Otherwise, jump to step (v).
7. Select the optimal (near-optimal) subset feature that has the highest fitness and highest number of the frequency in the run times.

3.1 Differential Evolution (DE)

One of the more potent evolutionary computation techniques, evolutionary algorithms (EAs) use concepts from biological evolution, such as mutation, crossover, and reproduction (selection), to find the answer to an optimization issue [10, 20]. In the past ten years, several EAs have proposed methods to identify the best way to classify spam emails. Additionally, Storm and Price developed the differential evolution (DE) method at Berkeley between 1994 and 1996. It is an evolutionary algorithm [21]. One type of EA that was suggested as a modified genetic algorithm (GA) is DE [19]. Comprehensive studies that are regularly published show that DE performs better than many other optimization techniques in terms of robustness and convergence speed over widely used benchmark functions[22]. In the commonly used benchmark issues, DE has demonstrated better performance than particle swarm optimization (PSO) and other EAs, and it requires less parameters [23]. One of the quickest optimization algorithms, DE is effective at resolving global optimization issues and only needs a few control parameters (CR, F, and NP) to produce better search results [24]. It is straightforward, has a strong optimization technique, improves convergence, and finds the actual global minimum independent of the initial values of the parameters [6]. It is a population-based algorithm that uses crossover, mutation, and selection—similar operators to genetic algorithms [14]. DE can handle non-differentiable, nonlinear, and multimodal objective functions and is simple to apply for optimizing problems [25]. By attempting to enhance a potential solution with respect to a specified quality metric, DE optimizes a problem. The individual (also known as a solution to the problem) is generated at random and is the first of several processes to be implemented in DE.

Iterations of crossover, mutation, and selection are then carried out until the optimal solution is obtained. For a third vector (person), known as the target vector, DE employed the differences between randomly chosen vectors (individuals) as the source of random variations. Weighted difference vectors are added to the target vector to provide trial solutions [26]. The target vector is altered during this procedure, which is known as the mutation operator. To create or increase the fitness of the parent individual, a recombination or crossover (discrete recombination) phase is used. The process of choosing individuals based on their fitness function, also known as the objective function for evolutionary algorithms, creates a new population. The fitness function establishes which of the solutions survival is. First, a continuous search space is used to randomly produce the individual, also known as a solution to the problem. The mutation, crossover, and selection processes were then repeated until the termination requirement was satisfied. With a few differences, the DE's methodology is comparable to that of genetic algorithms. The DE has been presented as a solution to the time-consuming nature of the various genetic algorithm variants that have been developed [14]. The mutation system that makes DE self-adaptive and the

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

selection process are the primary distinctions between GA and DE. Regardless of their fitness rating, every solution in the DE has an equal probability of being chosen as parents. Compared to genetic algorithms, DE has a higher chance of discovering a real global optimum function. Three fundamental evolution operators are used in the DE optimization process: mutation, crossover, and selection [23]. Creating an objective function is the first step in the conventional approach to an optimization issue. The optimization problem is typically defined as a minimization task by the objective function. It should be noted that DE has a few parameters, including the crossover constant (CR=rand (0,1)) and the mutation scaling factor (F=rand (0,2)), which are chosen by the practitioner together with the population size NP≥4 (it must be at least 4). Optimization performance may be significantly impacted by the selection of DE parameters F, CR, and NP. Greater diversity in the generated population is the result of a big value for the F parameter, whereas faster convergence is caused by a smaller value[27]. To find the subset space in a fair amount of time, we must employ heuristic techniques. Euclidean distance and cosine similarity are two novel fitness functions that we utilize in this work to choose subset characteristics. This reduces the dimension of the data, which can help algorithms run more quickly and efficiently. To get diverse findings, we also chose F=0.75, CR=0.50, and varied population sizes (NP), iterations, and runs for this investigation. Individuals are assessed for quality at each generation, and the best member is marked to monitor the evolution's progress. DE involves several phases, including:

3.1.1 Initialization:

A set of d-dimensional parameter vectors makes up a population at generation G (represented as $X_{i,G}$)), where each population vector denotes a potential solution to the issue (target vector). First, each person is created at random using a uniform probability distribution. Let's say we wish to choose the population size (number of vectors in the population) NP to optimize a function. The parameter vectors have the form:

$$X_{i,G} \quad i = 1,2, \ldots . \text{NP and G is the generation number}$$

Establish a starting goal for the vector population. Every target vector has a unique set of design parameters. For each parameter, a lower limit and an upper limit are defined ($X_i^l < X_{i,G} < X_i^u$). Furthermore, randomly select the initial parameter values uniformly on the intervals (X_i^l, X_i^u) and for each target vector, select other parameter vectors randomly.

3.1.2 Mutation

Five distinct learning mechanisms exist for the mutation in DE. The version DE/best/1/ bin, which is frequently used in various DE literatures and typically provides superior convergence, will be the one we employ in this paper. During the mutation stage, the DE algorithm generates new vectors by adding the weighted difference of two vectors to a third vector. For each target vector $X_{i,G} \quad i = 1,2, \ldots . NP$, a mutant vector for generation G+1 (denoted as,$V_{i,G+1}$) is generated as follows:

$$V_{i,G+1} = X_{best,G} + F(X_{r1,G} - X_{r2,G})$$

Where r_1, r_2 are random indices $r_1, r_2 \in \{1,2, \ldots, NP\}$[28]Error! Bookmark not defined.{Choi, 2021 #3817}; $r_1 \neq r_2 \neq i$ and $F \in [0,2]$ is a scale factor which controls the amplification of the differential variation$(X_{r2,G} - X_{r3,G})$.

To identify the best member of the population, several researchers began modifying the equation of mutation; this might result in a quicker convergence and better performance. Thus, the primary goal of the mutation operator is to increase the effective area of the search space that the algorithm takes into consideration by presenting some convergence in the population.

3.1.3 Crossover

Uniform crossover technique is used in DE, to increase the diversity and convergence of the perturbed parameter vectors, the trial vector $U_{i,G+1}$ is developed from the elements of the mutant vector $V_{i,G+1}$ and the target vector $X_{i,G}$.

$$U_{i,G+1} = (U_{1i,G+1}, U_{2i,G+1}, \ldots . , U_{Di,G+1})$$

Where

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

$$U_{ji,G+1} = \begin{cases} V_{ji,G+1} & if\ (rand(j) \le CR) or\ j = rn(i) \\ X_{ji,G} & otherwise \end{cases} \quad (1)$$

$$j = 1,2, \ldots . D$$

From Eq. (1), rand (j) is the j[th] evaluation of a uniform random number generator with outcome $\in [0,1]$. CR is crossover probability$\in [0,1]$. Crossover probability controls the fraction of parameter values that are copied from the mutant vector. A randomly chosen index $(rn(i) \in (1,2, \ldots, D)$ which ensures that $U_{i,G+1}$ gets at least one parameter from$V_{i,G+1}$. Therefore, the main goal of crossover is to increase the diversity in the population.

3.1.4 Selection

In selection operation, the trial vector at generation G+1 $(U_{i,G+1})$ is compared to the target vector at generation G$(X_{i,G})$. If the trial vector at generation G + 1 $(U_{i,G+1})$ yields more cost value than the target vector at generation G$(X_{i,G})$, then the trial vector replaces the target vector in the next generation (generation G + 1) (for maximization problem. Otherwise, the old value $X_{i,G}$ is retained and $X_{i,G+1}$ is determined as follows in Eq. (2):

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & if\ f\left(U_{i,G+1}\right) \ge f(X_{i,G}) \\ X_{i,G} & otherwise; \end{cases} \quad (2)$$

Where, f (.) denotes the cost function (fitness function) of the given vector.

3.2 Opposition-based learning (OBL)

Random generation is one of the primary problems with optimization techniques, according to some studies, and occasionally the opposite values are close to the ideal answer [12]. Machine learning approaches make up most strategies utilized to enhance the optimization techniques' search performance. This research uses opposition-based learning (OBL) to increase the convergence rate of DE by using opposite integers. To increase performance and choose the best subset of features, the opponent DE was using feature selection. EAs often begin with a set of initial solutions, or the initial population, and work to refine them to arrive at one or more optimal solutions. Starting with a random population, typically distributed uniformly throughout the whole search space, is a popular initialization when a priori information about the solution is not available. Numerous studies have been conducted in recent decades to enhance DE's performance. A recent development in computational intelligence, opposition-based learning (OBL) has been shown to be a useful idea for enhancing several optimization techniques [29]. Tizoshi et al. proposed OBL, a machine learning algorithm[30], The opposition-based differential algorithm is one of the machine learning algorithms that has included it [27] and a genetic algorithm based on antagonism. The OBL method was recently employed by a number of researchers to enhance the quality of DE [31]. The OBL was recently applied to different EAs. Rahnamayan et al. [12] for the first time utilized opposite numbers to speed up the convergence rate of an optimization algorithm. Rahnamayan et al established mathematical proofs and experimental evidence to verify the advantage of opposite points compared to additional random points [22]. The OBL was used for population initialization in their suggested opposition DE (ODE). According to the results, ODE outperformed DE at the maximum dimensionality. By computing the opposite value and the individual's fitness to its opposite, the OBL can be implemented. The search process is accelerated by keeping the fitter individual in the population. However, use the OBL to speed up the DE without causing it to stagnate or converge too quickly [19]. PSO and DE are examples of evolutionary computing (EC) techniques that use concepts from biological evolution to produce initial solutions (population) at random. When scanning space, these algorithms typically overlook the prior knowledge about these solutions. Because evolutionary processes are slow, evolutionary algorithms are computationally costly. Most optimization methods depend on population initialization; therefore, intelligent initialization techniques based on machine learning techniques are necessary for the optimal outcome since the best solution is chosen. As far as we are aware, this is the first time that opposing numbers have been used to enhance the DE when it comes to feature selection for the classification of spam emails. By exploiting random numbers and their opposites, OBL uses it to speed up differential evolution (DE). OBL has been used to speed up DE's rate of convergence. Without introducing any additional parameters, OBL was utilized to improve DE's performance. To enhance the rate of convergence in DE and examine the impact of dimensionality, opposite numbers have been employed. Additionally, the DE method is prone to convergence rate issues, which results in reduced accuracy when regulating its location and velocity. When starting a population and creating new populations during the evolutionary process, ODE employs opposite numbers.

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

The learning and search process can be accelerated by incorporating opposition-based learning into already-existing soft computing techniques. Enhanced DE with the use of OBL, which creates the opposite population from the original population and selects the finest attributes from the combined population. An algorithm may not find the target area or converge if it is not initialized optimally, which results in an efficient search for the intended answer.

OBL is used in DE to increase convergence speed, prevent parameter convergence, and increase fitness while choosing the key feature. The modification of the current approach is the primary distinction between this study's improvement of DE and the others. An optimization algorithm's rate of convergence has been accelerated by using opposite numbers. ODE is being used in this study to avoid parameter convergence and speed up the convergence of DE. The DE has OBL integrated into it.

3.3 The Material Used

The next subsections discuss how DE is configured. This presents more details about DE algorithms.

3.3.1 Control Parameters for DE

The control parameters of DE are the scale factor F, crossover CR, population size NP, number of iterations, ("No.of.iteration") and number of running ("No.of.run"). They are set as follows in Table 1

Table 1: The control parameters values used in DE.

Parameter Name	Parameter Value
CR	0.9
F	0.5
NP	100
No. of. Iteration	1000
No. of. Run	20

3.3.2 Fitness Function

Researchers employed the correlation coefficient (r) as the fitness function for both ODE schemes in this work. To choose the best (or nearly best) subset characteristics and to assist the SVM learning algorithm in increasing the email's classification accuracy, researchers utilize correlation coefficient functions as fitness functions in ODE algorithms. To increase its convergence capabilities, researchers have refined the traditional DE algorithms from their classical form. Our proposed function is Eq. (3).

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{(N\sum_i x^2 - (\sum_i x)^2)(N\sum_i y^2 - (\sum_i y)^2)}} \quad (3)$$

Where X_i is the feature value, $\bar{X}_i$ is mean value for the feature column, Y_j is the target value (output value) and $\bar{Y}_j$ is the target mean value. Many studies before having used the correlation coefficient to calculate the significance between the features without considering the output. The significance between the input (features) and the output (class label) is determined in this study using the correlation coefficient, and the significance result is compared to other results. A better fitness function is indicated by the highest significant value. Numerous researchers have previously employed DE algorithms with various fitness functions; however, none have examined the relationship between the input and the output or used significance as a fitness function.

3.3.3 The Modulator

In this paper, researchers use two different modulators. The first modulator is used for Opposition Differential Evolution (ODE). Furthermore, the main objective of this paper is to investigate the effect of modulation formulas on extracting optimally effective features.

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

In the ODE researchers employ a binary modulation formula presented by [32] as shown in the Eq. (4):

$$FS_i = \begin{cases} 1 & \text{if} \quad \text{rand}() >> \exp(-|x_i| \\ 0 & \text{if} \quad \text{rand}() < \exp(-|x_i| \end{cases} \qquad (4)$$

From the above Equation FS_i refers to the corresponding binary of the real value gene where $|x_i|$, $rand(\ )$ is a function that generates a real random number between 0 and 1, and $-(|x_i|)$ is an exponential value of the correspondent gene$|x_i|$. If $rand(\ )$ is greater than or equal to $\exp(-|x_i|)$ then $FS_i = 1$ else $FS_i = 0$. On the other hand, after transforming the value of gene $|x_i|$, using the modulation into binary code and if the value of $FS_i = 1$, then the corresponding feature is selected as an optimal feature. In contrast, the corresponding feature is not chosen and is not regarded as an ideal feature if the bit contains 0. To choose the best (near-best) subset of features, this study uses two datasets, each of which has many features.

4. Implementation, results, and discussion

This paper implements the suggested scheme (ODE-SVM) using the opposite definition. This demonstrates how the search space can be enhanced by the suggested OBL. To make comparisons easier, the features are arranged in descending order. The value of each attribute indicates its significance and impact on the email. For the second testing comparison procedure, the best (near-best) subset features were chosen as input. However, researchers ignore traits that are redundant, useless, or have a score near "0". Following the feature subset selection, testing was conducted once more. Because the degree of classification depends on the number of features extracted from emails, researchers discovered that the degree of classification accuracy has grown when compared to the initial testing. Consequently, classification accuracy increases when the irrelevant features are reduced, and vice versa. After calculating the similarity score, the datasets are used for crosschecking.

4.1 Results and Discussions

This paper's findings are divided into two sections. The F-measure and accuracy are calculated in the first section, and the correlation coefficient is used in the second to evaluate the degree of correlation between the system outcomes and others. The correlation coefficient, however, indicates that these outcomes have been statistically enhanced. Figure 2 illustrates the Gains charts for both training and testing results after selecting the optimal (or near-optimal) subset features based on SVM as a classifier and BDE as a feature subset selection. In the Gains charts with a baseline, the best line is ($Best-BDESVM), and the result of SVM after using BDE as feature subset selection is ($S- BDESVM).

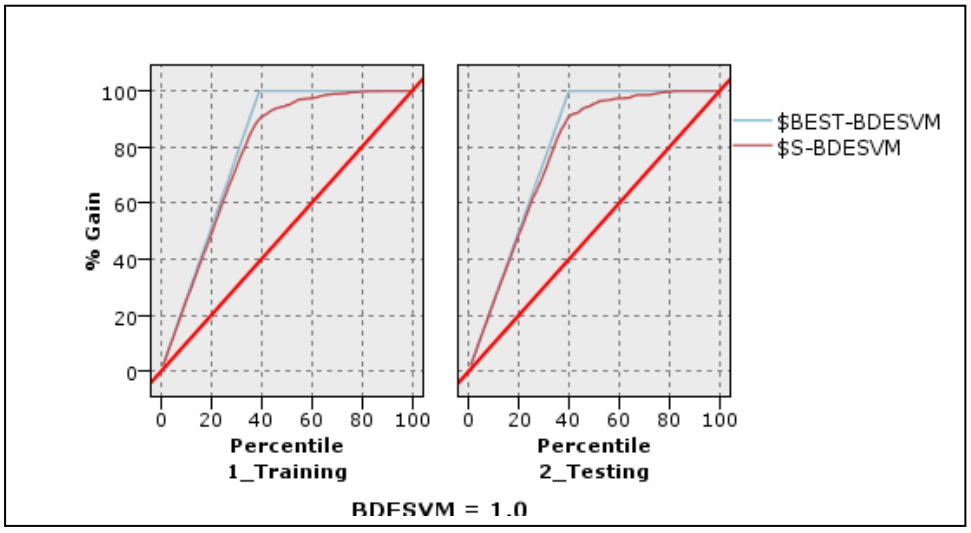

Figure 2. Training and testing results after using BDE

In this paper, two kinds of experiment were executed: ODE as a feature subset selection and SVM as a classifier. Figure 3 demonstrates the result example of using ODE as feature subset selection to select the optimal subset features based on population size=50, iteration no=1000 and number of runs=20 for "spambase" dataset and "spamassassin" dataset respectively.

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

Table 2: The example result for ODE of the best values for 20 runs on spambase dataset.

Rank	Features	Result
1	F21	20
2	F23	19
3	F7	18
.	.	.
.	.	.
21	F15	7
22	F24	6

Table 3: The example result for ODE of the best values for 20 runs on spamassassin dataset.

Rank	Features	Result
1	F46	20
2	F55	20
3	F19	20
.	.	.
.	.	.
.	.	.
21	F68	9
22	F12	9

The results of selecting the best subset features using ODE as feature subset selection are shown in Tables 2 and 3. Following the selection of the ideal (or nearly optimal) subset features using ODE as a feature subset selection and SVM as a classifier, Figure 3 shows the Gains charts for both training and testing outcomes. In the Gains charts with a baseline, the best line is ($Best-ODESVM), and the result of SVM after using ODE as feature subset selection is ($S- ODESVM).

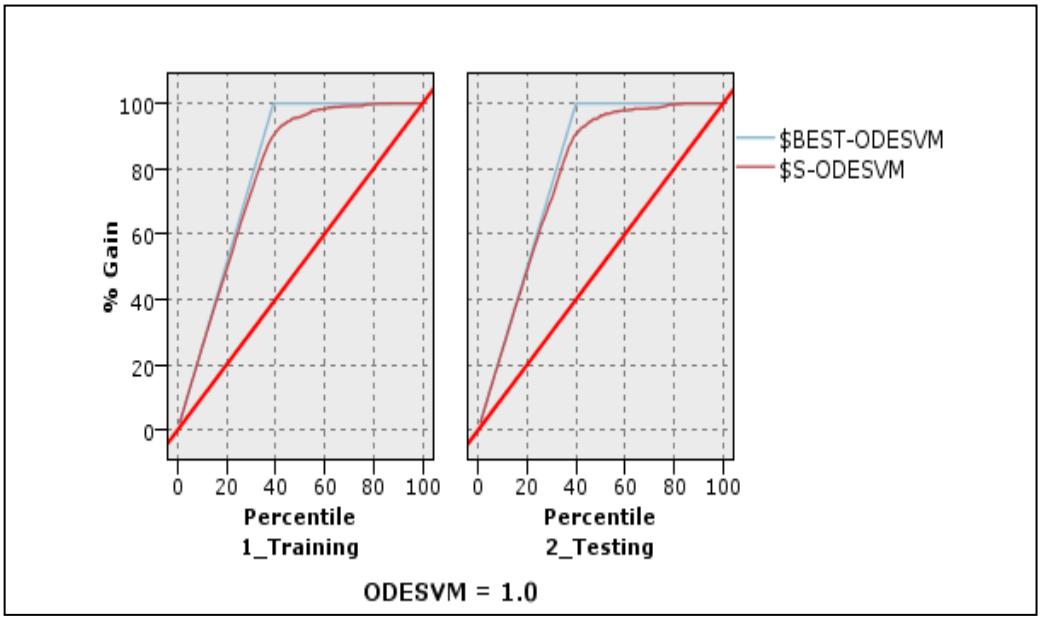

Figure 3. Training and testing result for SVM after using ODE.

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

With this technique, all characteristics depending on ODE are reduced in quantity. The classification accuracy results based on the SVM algorithm and ODE as FS are shown in Table 4, along with a comparison to the results based on BDE as FS. The classification accuracy using the SVM algorithm as a classifier and ODE as FS is 94.79%, while the accuracy using the SVM algorithm as a classifier and BDE as FS is 93.99% for testing, as shown in Table 4. To compare our results with those of other approaches, Table 5 shows the accuracy, recall, F-measure, and false positive outcomes.

Table 4: The classification accuracy result after using BDE and ODE with SVM as classifiers.

Classification	Accuracy %	Number of features
BDE-SVM	93.99	47
ODE-SVM	96.79	44

Table 4 demonstrates that, in comparison to the earlier findings, the accuracy result of emails classified using 44 features is improved. When compared to the findings, this accuracy shows that the majority of the emails were successfully sent to classification classifiers for recognition. As previously mentioned, the accuracy of the reduced feature subsets is assessed using a fitness function. This suggests that the suggested method increases the classification accuracy of e-mail spam. The accuracy, recall, and F-measure values are shown in the column of Table 5, while the percentages of each strategy are shown in the row. Additionally, Table 5 contrasts two types of experiments used in this paper: binary ODE-based feature subset selection and SVM-based classification. While the latter is in charge of classifying the chosen features for the e-mail spam classification problem, the former is in charge of obtaining the optimal (or nearly optimal) subset feature from all features. The ODE-SVM scheme's performance in classifying spam emails is tested and evaluated using the SVM algorithm. The ODE schemes provide the characteristics to the SVM algorithm. The findings show that ODE-SVM, one of the two enhanced schemes, outperforms DE-SVM and GA-SVM, two comparable optimization strategies, in terms of accuracy. Numerous writers have improved the accuracy of email spam classification by using GA as an example.

Table 5: Comparisons of Accuracy, Recall, and F-measure using different numbers of features based on SVM.

approach	Accuracy %	Recall%	F-measure
SVM	93.55	0.96	0.93
BDE-SVM	93.99	0.96	0.95
ODE-SVM	96.79	0.97	0.98

The classification accuracy results based on the SVM algorithm and various feature selection techniques are compared in Table 5. The accuracy of the suggested ODE-SVM schemes is 96.79% higher than that of utilizing SVM alone or BDE with SVM, according to the generalization of the results. The comparisons of correct results between the ODE-SVM result and the prior result are shown in Figure 4. The y-axis in Figure 4 shows the accuracy percentages for the suggested schemes using a column chart, while the x-axis compares the accuracy of our earlier schemes with the most recent plan. In comparison to the BDE-SVM scheme with selected characteristics or only utilizing SVM as a classifier with full features, Figure 4 demonstrates that the upgraded scheme (ODE-SVM) gets the highest accuracy.

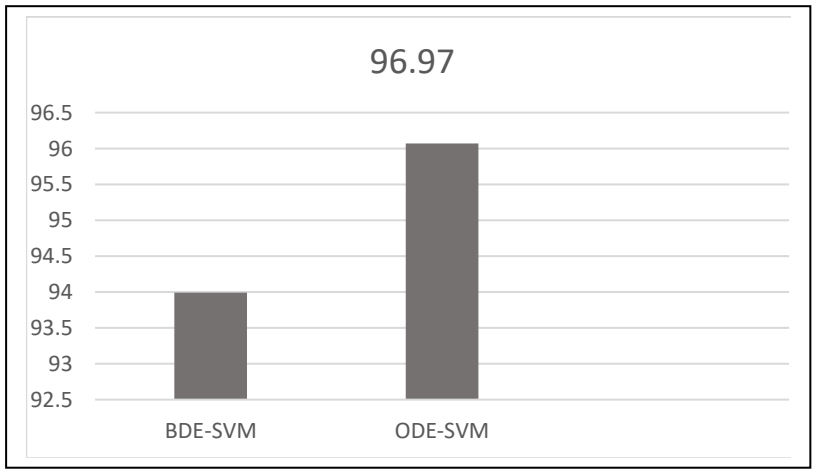

Figure 4. Comparisons among our proposed schemes

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

4.2 Compression with other Methods

A comparison between our suggested systems and other techniques to improve email spam classification is shown in this section. Our improvement scheme's accuracy result is compared to the findings of (El-Alfy et al., 2011; Idris and Selamat. 2014; Maldonado et al., 2013). Paper 2 begins with a discussion of these techniques. Because both approaches make use of the same datasets, researchers have chosen them for comparison. Researchers have discovered that our outcome is superior to the results of other methods when comparing them with ours.

Table 6: Comparison of our Accuracy with others

Method	Accuracy %
GMDH	91.70
PSO-NSA	91.22
BDE-SVM	93.99
ODE-SVM	96.79

The comparison between our technique and the outcomes of other approaches for classifying e-mail spam is shown in Table 6. Researchers observe that the suggested plan produces findings that are more accurate.

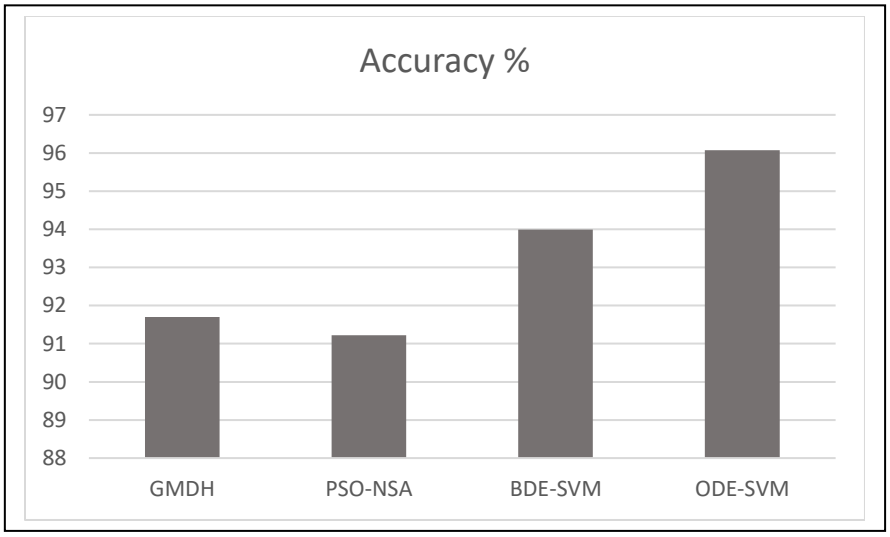

Figure 5. Comparisons among our proposed schemes

Figure 5 displays the outcomes derived from Table 6. The accuracy comparison charts for ODE-SVM results utilizing various column chart techniques are shown in Figure 5. Additionally, Figure 5 compares ODE-SVM's accuracy to that of other models. According to Figure 5, ODE-SVM produces more accurate results than BDE-SVM and other models.

5. Conclusion

In this work, we introduced a spam classification method based on the Opposition Differential Evolution (ODE) algorithm, a sophisticated optimization method modeled after biological opposition. The aim of this study was to improve email classification systems' ability to differentiate between spam and non-spam emails, hence tackling the increasing difficulties caused by spamming strategies. The selection of subset characteristics was the main emphasis of this research, and its contribution was to pick the subset features using binary opposition differential evolution (BODE). As a result, the difficulty is in offering a subset feature and classification technique, like SVM, for email spam categorization. One way to choose the subset characteristics to lower the higher dimension and improve accuracy is the method we have suggested. The outcomes of the suggested method are regarded as one of the keys/important study answers for the classification of email spam. To cut down on unsolicited messages, an email spam classification system might be useful. Our suggested method is assessed and contrasted with a few of the existing methods for classifying spam emails. Overall, the following benefits are provided by the suggested method: increased the classification precision of spam emails. This research presents a novel feature selection method based on the binary differential evolution optimization strategy. The suggested method's performance is

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

contrasted with that of existing population-based feature selection methods, such as GA and PSO. It is demonstrated that the suggested BDE used less memory than alternative techniques, resulting in a shorter execution time. Furthermore, the suggested method outperformed both GA and PSO in terms of classification performance when tested on an e-mail spam classification problem, with an accuracy of 93.99%. The efficiency of the suggested BDE technique was demonstrated by all the results. In the future, the author will choose the important traits using opposition particle swarm optimization.

Acknowledgment

The authors are thankful to the Deanship of Graduate Studies and Scientific Research at University of Bisha for supporting this work through the Fast-Track Research Support Program.

References

1. Hamed, N.O., A.H. Samak, and M.A. Ahmad, Cloud e-mail security: An accurate e-mail spam classification based on enhanced binary differential evolution (BDE) algorithm. Journal of Intelligent & Fuzzy Systems, 2021. 41(6): p. 5943-5955.
2. R.Deepa Lakshmi, N.R., Supervised Learning Approach for Spam Classification Analysis using Data Mining Tools (IJCSE) International Journal on Computer Science and Engineering 2010. Vol. 02, (No. 08): p. 2760-2766.
3. Saad, O., A. Darwish, and R. Faraj, A survey of machine learning techniques for Spam filtering. IJCSNS, 2012. 12(2): p. 66.
4. Salehi, S. and A. Selamat. Hybrid simple artificial immune system (SAIS) and particle swarm optimization (PSO) for spam detection. 2011. IEEE.
5. Suebsing, A. and N. Hiransakolwong, A Novel Technique for Feature Subset Selection Based on Cosine Similarity. Applied Mathematical Sciences, 2012. 6(133): p. 6627-6655.
6. Khushaba, R.N., A. Al-Ani, and A. Al-Jumaily. Differential evolution based feature subset selection. in Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. 2008. IEEE.
7. Lin, S.-W., et al., Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Systems with Applications, 2008. 35(4): p. 1817-1824.
8. Chen, J., et al., Feature selection for text classification with Naïve Bayes. Expert Systems with Applications, 2009. 36(3, Part 1): p. 5432-5435.
9. Choi, T.J., J. Togelius, and Y.-G. Cheong, A fast and efficient stochastic opposition-based learning for differential evolution in numerical optimization. Swarm and Evolutionary Computation, 2021. 60: p. 100768.
10. Qin, A. and X. Li. Differential evolution on the CEC-2013 single-objective continuous optimization testbed. in Evolutionary Computation (CEC), 2013 IEEE Congress on. 2013. IEEE.
11. Jun, Z., et al., Evolutionary Computation Meets Machine Learning: A Survey. Computational Intelligence Magazine, IEEE, 2011. 6(4): p. 68-75.
12. Rahnamayan, S., H.R. Tizhoosh, and M.M. Salama, Opposition-based differential evolution. Evolutionary Computation, IEEE Transactions on, 2008. 12(1): p. 64-79.
13. Tušar, T. and B. Filipič. Differential evolution versus genetic algorithms in multiobjective optimization. in Evolutionary Multi-Criterion Optimization. 2007. Springer.
14. GA, D.K. and S. Okdem, A simple and global optimization algorithm for engineering problems: differential evolution algorithm. Turk J Elec Engin, 2004. 12(1).
15. Sun, J., et al., Analysis of the distance between two classes for tuning SVM hyperparameters. Neural Networks, IEEE Transactions on, 2010. 21(2): p. 305-318.
16. Morariu, D., L. Vintan, and V. Tresp. Evolutionary feature selection for text documents using the SVM. 2006.
17. Fagbola Temitayo, O.S., digun Abimbola Hybrid GA-SVM for Efficient Feature Selection in E-mail Classification Computer Engineering and Intelligent Systems 2012. Vol 3, No.3, 2012(No.3,): p. 17-28.
18. Al-Tashi, Q., et al., Approaches to multi-objective feature selection: A systematic literature review. IEEE Access, 2020. 8: p. 125076-125096.
19. Ahandani, M.A. and H. Alavi-Rad, Opposition-based learning in the shuffled differential evolution algorithm. Soft computing, 2012. 16(8): p. 1303-1337.
20. Kokash, N., An introduction to heuristic algorithms. Department of Informatics and Telecommunications, 2005.
21. Jia, D., X. Duan, and M.K. Khan, An Efficient Binary Differential Evolution with Parameter Adaptation. International Journal of Computational Intelligence Systems, 2013. 6(2): p. 328-336.

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024

22. Rahnamayan, S., H.R. Tizhoosh, and M. Salama, Opposition versus randomness in soft computing techniques. Applied Soft Computing, 2008. 8(2): p. 906-918.
23. Zhang, X., et al., Multi-class support vector machine optimized by inter-cluster distance and self-adaptive deferential evolution. Applied Mathematics and Computation, 2012. 218(9): p. 4973-4987.
24. Abraham, A., S. Das, and A. Konar. Document Clustering Using Differential Evolution. in Evolutionary Computation, 2006. CEC 2006. IEEE Congress on. 2006.
25. Storn, R. and K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 1997. 11(4): p. 341-359.
26. Omran, M.G., Using opposition-based learning with particle swarm optimization and barebones differential evolution. Particle Swarm Optimization, InTech Education and Publishing, 2009.
27. Rahnamayan, S., H.R. Tizhoosh, and M.M. Salama. Opposition-based differential evolution (ODE) with variable jumping rate. in Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium on. 2007. IEEE.
28. Wang, H., et al., Enhancing particle swarm optimization using generalized opposition-based learning. Information Sciences, 2011. 181(20): p. 4699-4714.
39. Tizhoosh, H.R. Opposition-based learning: a new scheme for machine intelligence. in Computational intelligence for modelling, control and automation, 2005 and international conference on intelligent agents, web technologies and internet commerce, international conference on. 2005. IEEE.
30. Xingshi, H., et al. Feature Selection with Discrete Binary Differential Evolution. in International Conference on Artificial Intelligence and Computational Intelligence, 2009. AICI '09. . 2009.

DOI: https://doi.org/10.54216/FPA.170229
Received: February 24, 2024 Revised: May 22, 2024 Accepted: October 26, 2024