A Novel Bio-inspired Hybrid Metaheuristic for Unsolicited Bulk Email Detection

Tushaar Gangavarapu[1][0000−0002−0489−9573] and Jaidhar C.D.[2]

[1] Automated Quality Assurance (AQuA) Machine Learning Research, Content Experience and Quality Algorithms, Amazon.com, Inc.
tusgan@amazon.com
[2] Dept. of Information Technology, National Institute of Technology Karnataka

Abstract. With the recent influx of technology, Unsolicited Bulk Emails (UBEs) have become a potential problem, leaving computer users and organizations at the risk of brand, data, and financial loss. In this paper, we present a novel bio-inspired hybrid parallel optimization algorithm (Cuckoo-Firefly-GR), which combines Genetic Replacement (GR) of low fitness individuals with a hybrid of Cuckoo Search (CS) and Firefly (FA) optimizations. Cuckoo-Firefly-GR not only employs the random walk in CS, but also uses mechanisms in FA to generate and select fitter individuals. The content and behavior-based features of emails used in the existing works, along with Doc2Vec features of the email body are employed to extract the syntactic and semantic information in the emails. By establishing an optimal balance between intensification and diversification, and reaching global optimization using two metaheuristics, we argue that the proposed algorithm significantly improves the performance of UBE detection, by selecting the most discriminative feature subspace. This study presents significant observations from the extensive evaluations on UBE corpora of 3,844 emails, that underline the efficiency and superiority of our proposed Cuckoo-Firefly-GR over the base optimizations (Cuckoo-GR and Firefly-GR), dense autoencoders, recurrent neural autoencoders, and several state-of-the-art methods. Furthermore, the instructive feature subset obtained using the proposed Cuckoo-Firefly-GR, when classified using a dense neural model, achieved an accuracy of 99%.

Keywords: Evolutionary computing · Feature selection · Internet security · Metaheuristics · Natural language processing · Phishing · Spam

1 Introduction

In recent years, due to the increased ease of communication via emails, Unsolicited Bulk Emails (UBEs) have become a common problem. UBEs can be majorly divided into two related categories, i.e., spam and phishing emails. Spam emails constitute the category of bulk emails sent without users' consent, primarily with the intent of marketing (e.g., diet supplements, unlicensed medicines, etc.). Phishing is a more severe type of semantic attack aimed at deceiving users into providing sensitive information such as bank details, account numbers, and

others. According to the internet security threat report [15], 55% of the emails constituted spam in 2017 (2% more than in 2015-16). Gartner study in the United States showed that approximately 109 million adults received phishing email attacks which resulted in an average loss of $\$1,244$ per victim.

Evidently, spam and phishing rates are proliferating and the effects of such UBEs include theft of user identities, intellectual properties, degradation of mailing efficiency and recipient's productivity. Automatically detecting such UBEs has become a prominent area of research and hence has drawn a variety of considerations from researchers including behavior-based [20,16] and content-based [5] anti-UBE methods. In spite of the continuous efforts to avoid UBEs, attackers continuously change their strategies of executing spam and phishing attacks which makes it crucial to develop UBE detection methods with high performance. Furthermore, most of the UBEs are very similar to ham emails, making it extremely challenging to curb UBE attacks purely based on the email content. Moreover, most of the current email filtering approaches are static and can be defeated by modifying the link strings and email content. In this study, we mine forty content and behavior-based features of emails from the existing literature along with 200 Doc2Vec features of the email body content, to extract the syntactic and semantic information embedded in the emails.

Many attempts to detect and classify UBEs have been made [7]. These methods include white and blacklisting, content and network-based filtering, client-side toolbars, server-side filters, firewalls, and user awareness [19]. Usually, the data contained in the emails is very complex and multi-dimensional, resulting in higher time and space complexity, and low classifier performance [3]. Thus the cost of computation can be reduced while increasing the classification performance with a discriminative and informative feature subset. Dimensionality reduction to aid the classification of UBEs can be performed either by feature selection or feature extraction. This study focuses on a bio-inspired hybrid feature selection approach to discriminate between the email types.

A metaheuristic aims at generating or selecting a low-level heuristic which might provide a better solution than classical approaches to an optimization problem. The success of metaheuristics can be attributed to the nature of swarm intelligence algorithms being flexible and versatile, in the sense that they mimic the best features in nature [8]. Recently, Cuckoo Search (CS) [18] and Firefly Algorithm (FA) [17] have gained popularity from many researchers—their performance proved to be more efficient in solving global optimization problems than other metaheuristics [14,6]. CS was inspired by the obligate brood parasitism of certain species of the cuckoo bird by laying their eggs in the nests of other species birds, in combination with their Lévy flight behavior to search for food. FA was inspired by the observations of the flashing patterns and practices of fireflies who attract their partners using intensity of the emitted light.

This paper presents a novel hybrid bio-inspired metaheuristic called the Cuckoo-Firefly-GR to select the most discriminative and informative feature subset from a set of both content-based and behavior-based features needed to classify UBEs. The hybrid metaheuristic combines the evolutionary natures of

CS and FA using the concepts of random walk in CS and mechanisms of FA to generate or select fitter individuals. Furthermore, the hybrid metaheuristic is combined with a Genetic Replacement (GR) of low fitness individuals. The novelty of our hybrid metaheuristic lies in the way that the abandoned nests in CS and low fitness fireflies in FA are genetically replaced, and in the way that combines the complementary strengths and advantages of Cuckoo-GR and Firefly-GR. To the best of our knowledge, the existing literature that combines CS or FA, and the Genetic Algorithm (GA) modifies the non-abandoned nests using GA. Moreover, the previously available methods that combine CS and FA do not use GR. The goal of the proposed hybrid metaheuristic is to produce more accurate feature subset and globally optimize the task of feature selection, by reducing the average number of fitness evaluations, in turn, improving the selection performance. The experimental results emphasize the superiority of the proposed hybrid metaheuristic over the base optimizations (Cuckoo-GR and Firefly-GR), dense autoencoders, and Long Short Term Memory (LSTM) autoencoders. The key contributions of this paper can be summarized as:

- Leveraging vector space content modeling, to extract the syntactic and semantic relationships between the textual features of the email body.
- Design of a hybrid metaheuristic based on the evolutionary natures of CS and FA that uses GR to establish an optimal balance between intensification and diversification in the population.
- Evaluating the effectiveness of the proposed hybrid metaheuristic (Cuckoo-Firefly-GR) in the classification of UBEs. Our results indicate the efficacy of the proposed metaheuristic over various state-of-the-art methods.

The rest of this paper is organized as follows: Section 2 reviews the relevant aspects of CS, FA, and GA algorithms. Section 3 elucidates the approach followed to extract content-based and behavior-based features, and presents the proposed metaheuristic. Results of the proposed metaheuristic on ham, spam, and phishing corpora are presented and discussed in Section 4. Finally, Section 5 concludes this study and presents future research directions.

2 Cuckoo Search, Firefly, and Genetic Algorithms

In this section, we review the relevant aspects of CS, FA, and GA used in designing the hybrid metaheuristic (Cuckoo-Firefly-GR).

2.1 Cuckoo Search Algorithm

Yang and Deb [18] developed CS, a metaheuristic search algorithm to efficiently solve the global optimization problems. Existing body of literature reports that CS is far more efficient than many other metaheuristic search approaches including particle swarm optimization and GA. CS has since been successfully applied to various fields including biomedical engineering, antenna design, power systems, and microwave applications [14].

CS is primarily inspired by the Lévy flight behavior of the cuckoo birds while searching for food, along with their aggressive reproduction strategy. Generally, cuckoo birds do not build their nests; instead, they lay their eggs in communal nests so that surrogate parents unwittingly raise the cuckoo brood. Moreover, cuckoo birds may remove host bird's eggs to increase the hatching probability of their eggs. The host bird can build a new nest at a new location or throw away the cuckoo eggs if it finds that the eggs are not its own. The following three idealized rules are employed in the design of CS using the breeding analogy:

- Each cuckoo lays one egg at a time and deposits the egg in a randomly chosen nest among the available communal nests.
- A certain number of best nests with high-quality eggs will be carried on to the subsequent generations, thus ensuring that good solutions are preserved.
- There are a fixed number of nests, and the probability of a host bird discovering a cuckoo egg is $p_a \in [0, 1]$. When the host bird encounters a cuckoo egg, it can either build a new nest at a new location or throw away the egg.

In an optimization problem, every egg in a communal nest represents a possible solution, and a cuckoo egg constitutes a new solution candidate $\boldsymbol{x}^{(t+1)} = (x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_d^{(t+1)})^{\mathrm{T}}$. For each iteration, a cuckoo egg is randomly selected to generate new solutions. This random search can be executed efficiently using a Lévy flight. The Lévy flights are a type of random walks where the step lengths follow a specific heavy-tailed probabilistic distribution and the step directions are random and isotropic. Since Lévy flights have infinite mean and variance, some solutions will be closer to the current best solutions and others will be placed away from the best solutions, thus enabling a broader search space exploration. For a given cuckoo, say c, the Lévy flight on the current solution $x_c^{(t)}$ generates the new solution parameter $x_c^{(t+1)}$ which is computed as $x_c^{(t)} + \alpha \oplus \text{Lévy}(\lambda)$, where $\oplus$ indicates entry-wise multiplication and α is a positive constant, scaled using the dimensions of the search space. The value of α determines how far a particle can move by a random walk, in a fixed number of cycles. The computation of $x_c^{(t+1)}$ is a Markov chain, which is a stochastic equation for a random walk and the new solution is only reliant on: the current solution $(x_c^{(t)})$ and the probability of transition. The probability of transition is modulated using the Lévy distribution as: $\text{Lévy}(\lambda) \sim u = t^{-\lambda}$, where λ defines the decay of the probability density function with t. For most cases, $\alpha = 0.01$ and $1 < \lambda \leq 3$ [18]. This study employs CS with $N = 25$ nests, $\alpha = 0.01$, $\lambda = 2.5$, and $p_a = 0.4$, for a maximum of 50 cycles. In nature, many insects and animals often follow the properties of Lévy flights while searching for food in a random or quasi-random manner [14]. Using CS via Lévy flights helps to explore the search space more effectively when compared to algorithms using standard Gaussian process, by avoiding the problem of being trapped around local optima.

2.2 Firefly Algorithm

Yang developed FA [17] based on the flashing patterns and social behaviors of fireflies. Fireflies produce luminescent flashes by process of bioluminescence, as

a signal system to attract mating partners and potential prey. The rhythmic flashing, rate of flashing, and the amount of time form part of the signaling system that brings both sexes together. Owing to the effectiveness of FA in solving global optimization problems, it has been applied to various fields including stock forecasting, structure design, and production scheduling [6]. The existing literature corroborates that, although CS has outperformed FA in multimodal optimization, FA is better at generating optimum or near-optimum value in limited time [2]. The following three idealized rules are employed in the development of FA using the firefly signaling analogy:

- All the fireflies are unisexual, and every individual firefly will be attracted to every other firefly regardless of their sex.
- The attractiveness is proportional to the brightness, and they both decrease as the distance increases. Thus, for any two flashing fireflies, the less bright firefly will move towards the brighter one. Also, a firefly will move randomly, if there is no brighter firefly.
- The light intensity or brightness of a firefly is associated with the landscape of the objective function to be optimized.

There are two main concerns in FA: the variation of the light intensity and formulation of the attractiveness among fireflies. For simplicity, it can be assumed that the attractiveness is determined by the light intensity or brightness of a firefly, which is in turn dependent on the objective function ($f(x)$). For a maximization problem, the light intensity $I_i(x)$ of a particular firefly, say i, at a location, say x, can be chosen such that $I_i(x) \propto f(x)$, $\forall i$. However, the attractiveness (β) is relative and is judged by other fireflies, thus varies with the distance between the fireflies (r). Since the brightness decreases with an increase in the distance and the flashing light is also absorbed in the media, attractiveness must be modeled using both these parameters. In the simplest form, the brightness $I(r)$ with I' source brightness follows the inverse square law, $I(r) = I'/r^2$ However, in a given medium with γ light absorption coefficient, $I(r)$ varies monotonically and exponentially as: $I(r) = I' \cdot exp(-\gamma r)$. Since the attractiveness of a firefly is proportional to the brightness seen by the adjacent fireflies, β can be computed as: $\beta = \beta' \cdot exp(-\gamma r^2)$, where β' is the attractiveness when $r = 0$. The distance between any two fireflies, say i and j at positions x_i and x_j, can be computed as the Cartesian between them, i.e., $r_{i,j} = ||x_i - x_j||_2$ or the l_2-norm. The movement of a less bright firefly, say i, towards a brighter (more attractive) firefly, say j, is determined using $x_i = x_i + \beta' \cdot exp(-\gamma r_{i,j}^2) \cdot (x_j - x_i) + \alpha \cdot \mathcal{G}_i$, where the second term accounts for attraction while the third term is randomization with a vector of random variables $\mathcal{G}_i$ drawn from a Gaussian distribution. In most cases, $\gamma = 1$, $\beta' = 1$, and $\alpha \in [0, 1]$. This study employs FA with $N = 25$ fireflies, $\gamma = 1$, $\alpha \in [0.1, 1]$, and $\beta' = 1$, for a maximum of 50 cycles.

2.3 Genetic Algorithm

GA is a classical optimization algorithm inspired by Darwin's theory of evolution, natural selection process, and various genetic operators such as crossover and

mutation. GA has been successfully applied to various fields, including job-shop scheduling, rule set prediction, feature selection, and others.

To solve an optimization problem, GA constructs a random population of individual solutions. Subsequently, the fitness function, i.e., the objective function to be optimized measures the quality of an individual in the current population. Bio-inspired operators including selection, crossover, and mutation aid in the conversion of one generation to the next generation. First, the reproduction or selection procedure such as roulette-wheel, truncation, or tournament, is executed to select a potential set of parents based on their fitness scores. The crossover and mutation follow the selection procedure, aiding in the construction of the subsequent generation. Computationally, the variables are denoted as genes in a chromosome and are codified using a string of bits. These parent bit sequences are selected randomly based on their fitness scores, to produce the next generation of individuals. Several crossovers are performed with a crossover probability p_c, to replace the population and replicate the randomness involved in an evolutionary process. Finally, certain bits in the newly formed individuals are swept and changed with a small mutation probability p_m. The quality of the new generation of chromosomes is assessed using the fitness function. This procedure is repeated until the fitness values of the generated solutions converge.

3 Proposed Methodology

First, we present the procedure utilized to extract features from emails, followed by the mathematical formulation of the optimization problem. Then, we describe the proposed hybrid metaheuristic employed to facilitate selection of an informative feature subspace. Finally, we discuss the Multi-Layer Perceptron (MLP) model used in the UBE detection. The employed pipeline is presented in Fig. 1.

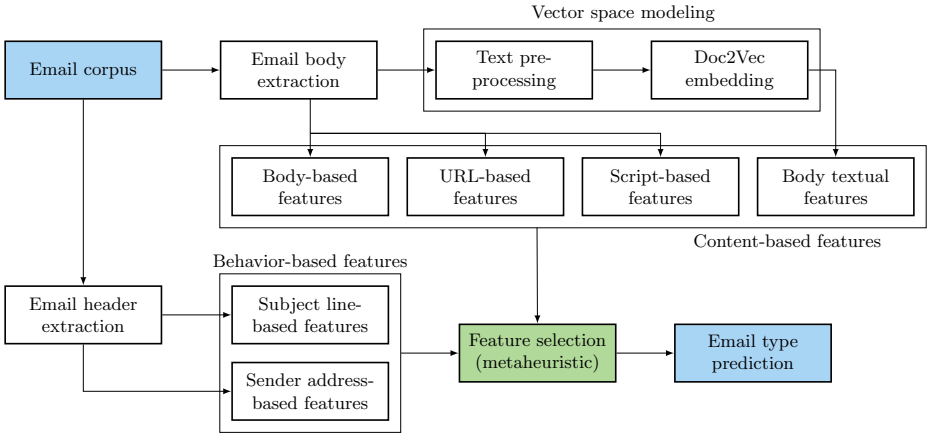

Fig. 1. Pipeline used to efficiently predict the email type using a hybrid of content-based and behavior-based email features.

3.1 Extraction of Content-based and Behavior-based Features

The features of emails considered in this study are internal to emails, rather than those from external sources like domain registry information or search engine information. Existing literature has shown that internal features form a comparatively more informative feature subset, as most of the external data such as DNS information changes regularly [16]. This study considers forty potential content and behavior-based features used in the existing literature along with 200 Doc2Vec features extracted purely from the message content.

Every email consists of two parts: email header and email body. The email header consists of general identification information including subject, date, from, to, and the route information followed by the email transfer agent. Forty significant features extracted from the email messages can be roughly categorized into five major categories: subject line-based features, sender address-based features, body-based features, URL-based features, and script-based features.

The subject line-based features include boolean flags to check if the email is a reply or forwarded email, presence of words like *verify*, *debit*, or *bank*, along with numeric features including word count, character count, and subject richness. The richness of a given text is computed as the ratio between number of words and number of characters. The sender address is mined for features including word count and character count, in addition to boolean checks aimed at detecting if the sender's domain, email's modal domain, and the reply-to domain are different. The subject line-based features and sender address-based features model the behavioral aspects of an email.

The body-based features include boolean flags indicating the presence or absence of HTML tags, forms, words like *suspension*, and phrases like *verify your account*, along with numeric attributes such as word count, character count, distinct word count, function word (e.g., *account*, *identity*, etc.) count, and body richness. The features extracted from the URLs in the email body include continuous attributes like total IP address count, total URL count, internal and external link count, image link count, total domain count, port count, and highest period count. In addition to these numeric features, certain boolean flags are also used to check for the presence of @ symbol in URLs, words such as *login*, *click*, *here*, or *update* in the link text, links mapping to a non-modal domain, URLs accessing ports other than 80, and IP addresses rather than domain names. Finally, the script-based features include boolean flags to check for the presence of scripts, JavaScript, and non-modal external JavaScript forms, and to check if the script overwrites the status bar in the email client. Furthermore, the number of on-click events are also measured as a part of the script-based features. The body-based features, URL-based features, and script-based features constitute the content-based aspects of an email.

In addition to these forty potential features of an email, Doc2Vec embedding of the email message content was performed to capture the syntax and semantics of the UBE's content. Various preprocessing steps including tokenization, stopword removal, and stemming/lemmatization were performed to achieve text normalization. First, punctuation marks, URLs, multiple spaces, and special

characters were removed. Then, the resultant text is split into several smaller tokens during text tokenization. Most frequently occurring words (stopwords) were removed using the NLTK English stopword corpus. Furthermore, character casing and space trimming were performed to achieve normalized data representations. Finally, stemming was performed to achieve suffix stripping, followed by lemmatization, to convert the stripped words into their base forms. To avoid overfitting and lower the computational complexity, the words occurring in less than ten emails were removed. The preprocessed text is then modeled using Doc2Vec, to derive optimal data representations.

Doc2Vec or Paragraph Vectors (PVs) efficiently learn the term representations in a data-driven manner [11]. Doc2Vec embeddings aim at capturing the intuition that semantically similar email contents will have near-identical vector representations (e.g., *debit* and *bank*). Doc2Vec is essentially a neural network with a shallow hidden layer that facilitates the model to learn the distributed representations of paragraphs to provide a content-related assessment. The implementations available in Python Gensim package were used to extract the Doc2Vec style features. This study employs a PV distributed memory variant of the Doc2Vec, with a dimension size of 200 (trained for 25 epochs) due to its ability to preserve the word order in the email content. In this study, we used a combination of these 200 features along with forty potential features, resulting in a total of 240 features.

3.2 Problem Formulation

Feature selection aims at projecting a set of points from the $m-$dimensional (original) space to a smaller $k-$dimensional space with a substantial reduction of noise and minimal loss of information. When such a projection specifies a subset of the available dimensions, the feature selection problem has a combinatorial nature. Let $\mathcal{E}$ denote the set of n emails, indexed by e. Each email has a set of m features that define the email, $\mathit{\Pi}^{(e)} = \{\mathcal{F}_f^{(e)}\}_{f=1}^{m}$, with each feature ($\mathcal{F}$), indexed by f. Our ultimate goal is to learn a function (g) that estimates the probability of an email belonging to one of the email types (ham, spam, or phishing), given its features: $g(\mathit{\Pi}^{(e)}) \approx Pr(\text{email type} \mid \mathit{\Pi}^{(e)})$. However, the complexity and high dimensionality of $\mathit{\Pi}^{(e)}$ often make it challenging to train and generalize a classifier. Furthermore, the nature of $\mathcal{E}$ makes it cost and time intensive to learn g as a mapping from emails to probabilities. Thus we need a transformation (T) from $\mathit{\Pi}^{(e)}$ into a machine processable and easier-to-use form, $T : \mathit{\Pi}^{(e)} \longrightarrow \mathbb{R}^k$. Usually, $k \ll m$ is used to achieve dimensionality reduction. Now, the transformed data, $\pi^{(e)} = T(\mathit{\Pi}^{(e)})$, $\pi^{(e)} \in \mathbb{R}^k$ is used generalize g and make the problem more tractable: $g(\pi^{(e)}) = g(T(\mathit{\Pi}^{(e)})) \approx Pr(\text{email type} \mid \mathit{\Pi}^{(e)})$. Now, the problem is decomposed into two steps: estimating T using a feature selection approach and estimating g using $\{(\pi^{(e)}, \text{email type}^{(e)})\}_{e \in \mathcal{E}_{\text{train}}}$.

The first step is formulated as determining a feature subset from the given $m-$features resulting in the best discriminating capabilities. Thus, this step involves an optimization problem in the search space of all the possible feature subsets whose criterion function is the accuracy ($\mathit{\Phi}_c$) obtained with a classifier

(c). This study poses the selection of an optimal subset of features as an optimization problem such that: $\arg\max_{\mathcal{S}} \ \Phi_c(\mathcal{S}, \{\pi^{(e)}\}_{e\in\mathcal{E}_{\text{train}}}, \{\pi^{(e)}\}_{e\in\mathcal{E}_{\text{test}}})$, where $\mathcal{S}$ denotes the feature subset and $\pi^{(e)} = \{\mathcal{F}_f^{(e)}\}_{f=1}^{|\mathcal{S}|}$ is obtained by projecting $\Pi^{(e)}$ onto the subspace spanned by features in $\mathcal{S}$. In optimization problems, often a coding scheme is needed to transform the selected subset of features into a string form. In this study, every individual solution is represented as a binary string of $m-$bits where the f^{th} bit corresponds to f^{th} feature. We use a bijection as a coding scheme, i.e., a feature is only included if its corresponding bit value is set. The classification accuracy is used to compute the fitness values of each solution in the population. This study uses a random forest classifier with 100 classification and regression trees of maximum depth 2, as the fitness function, evaluated as 3-fold cross-validation on the training set. Furthermore, this study uses a classifier that can learn and generalize from the informative feature subset obtained from transformation T, as the function g.

3.3 Genetic Replacement of Low Fitness Individuals

Intensification aims at measuring the ability to exploit the local neighborhood of the existing solutions needed to improve them. Diversification relates to the ability to explore the global search space to generate diverse solutions. An optimal balance between intensification and diversification is crucial in attaining better performance using global heuristic search methods [12].

Both CS and FA algorithms provide intensification and diversification. The diversification in CS and FA is rendered by the random initialization of population, intensity and attractiveness (in FA), and Lévy flights (in CS). The proposed hybrid metaheuristic manages the abandoned nests more effectively by replacing a part of them genetically, using bio-inspired operators including crossover and mutation. Such replacement aids in the effective convergence of solutions within a limited number of cycles by reducing the total number of iterations [14]. We define p_g as the probability of genetic replacement such that: $N_a = p_a \times N$; $N_g = p_g \times N_a$, where N is the population size and p_a is the probability of abandonment. The individual solutions for genetic replacement are randomly selected from the set of abandoned solutions. The N_g solutions are generated using the parents selected from the set of better fitness individuals using a roulette-wheel selection. The genetic replacement of abandoned solutions enhances the balance between diversification and intensification via genetic mutation. In this study, we use $p_a = 0.4$, $p_g = 0.6$, and $p_m = 0.025$.

3.4 Proposed Cuckoo-Firefly-GR Hybrid Metaheuristic

Both CS and FA have their advantages and are successful in solving a wide range of optimization problems. These metaheuristics are further diversified using the concepts of genetic algorithm, aimed at replacing low fitness individuals. We propose a hybrid of these two metaheuristic optimizations called the Cuckoo-Firefly-GR, to facilitate the use of random walks and Lévy flights in CS in addition to the mechanisms of attractiveness and intensity in FA.

Algorithm 1: Proposed Cuckoo-Firefly-GR hybrid metaheuristic.

```
Divide the population into two diverse groups, say P1 and P2 each of size N
Random initialization of the populations P1 and P2
Evaluate the fitness of each individual solution
while (t < max generation) do
    do in parallel
        Perform CS with GR (using pa and pg) on P1
        Perform FA with GR (using pa and pg) on P2
    Rank solutions and find the current global best solution among P1 and P2
    Mix P1 and P2 and randomly shuffle the entire population
    Regroup the entire population into equisized diverse groups, P1 and P2
    Evaluate the fitness of each individual solution
Postprocess the results
```

Existing literature indicates that FA could automatically subdivide the entire population into subgroups based on attractiveness via light intensity variations [6]. Such highlights of FA, when combined with the reproductive behavior of CS, and enhanced with GR, provide exploration and diversification needed to obtain optimal solutions with faster convergence. We employ a parallel hybridization, wherein the location information of the current best solution is mixed and re-grouped, instead of re-finding new positions using a random walk. Such parallelization ensures the search in the optimal location of the previous cycle, rather than having to re-search randomly.

The pseudocode in Algorithm 1 summarizes the procedure followed in the Cuckoo-Firefly-GR parallel hybridization. This study employs a population size (N) of 25, for a maximum of 25 cycles.

3.5 Classification of Unsolicited Bulk Emails

This study uses an MLP model to facilitate the UBE classification. MLP offers several advantages including fault tolerance, adaptive learning, and parallelism. Above all, MLP learns distributed data representations, enabling generalization to new combinations of values of features, beyond those seen while training.

MLP is a feed-forward neural network with an input layer, one or more non-linear hidden layers, and one prediction layer. The first layer takes the optimal email features ($\mathcal{I}$) as the input, and the output of each layer is used to activate the input in the subsequent layer. The transformation at each layer l is given as: $\mathcal{I}^{(l+1)} = f^{(l)}(\mathcal{I}^{(l)}) = g^{(l)}(W^{(l)} \cdot \mathcal{I}^{(l)} + b^{(l)})$, where $W^{(l)}$ and $b^{(l)}$ are the weight matrix and bias at layer l, and $g^{(l)}$ is a non-linear activation function such as ReLU, logistic sigmoid, or tanh function.

MLP uses the backpropagation algorithm to compute the gradient of the loss function needed to learn an optimal set of biases and weights. In this study, we aim at optimizing the cross-entropy prediction loss using an MLP model with one hidden layer of 75 nodes with a ReLU activation function.

Table 1. Statistics of the datasets used in UBE classification.

Dataset	Components	#Samples	#Classes
$D_{h,s}$	H and S	3,051	2
$D_{h,p}$	H and P	3,344	2
$D_{h,s,p}$	H, S, and P	3,844	3

H: *Ham*, S: *Spam*, and P: *Phishing*.

4 Experimental Validation and Discussion

This section describes the datasets used, the conducted experiments, and the obtained results concerning the classification of emails. Finally, we benchmark against the existing state-of-the-art methods in the field of UBE classification.

4.1 Raw Email Corpus

The raw email corpus used in this study consists of 3,844 emails comprising of 2,551 ham emails (66%), 793 phishing emails (21%), and 500 spam emails (13%), obtained from [16]. From these emails, three datasets were created. The first dataset was used to investigate the effectiveness of the proposed approach in spam classification: the dataset combined ham and spam emails from the corpus. The second dataset combined ham and phishing emails and aimed at testing the feature selection by the hybrid metaheuristic. To account for the fact that real-world email system could simultaneously receive ham, spam, and phishing emails, we created the third dataset comprising of all the emails. The statistics of these datasets are tabulated in Table 1.

Table 2. Comparison of the performance (%) of the hybrid metaheuristic over various feature selection approaches.

Approach	Metric	Dataset		
		$\mathbf{D_{h,s}}$	$\mathbf{D_{h,p}}$	$\mathbf{D_{h,s,p}}$
Cuckoo-Firefly-GR	ACC	**99.78**	**99.40**	98.74
	MCC	**99.20**	**98.35**	**98.10**
Cuckoo-GR	ACC	98.47	97.11	**98.79**
	MCC	94.34	91.97	97.57
Firefly-GR	ACC	97.37	97.61	92.36
	MCC	92.73	93.36	94.16
Deep Autoencoders (4 Layers)	ACC	89.85	95.62	83.54
	MCC	57.82	87.65	65.92
Deep Autoencoders (8 Layers)	ACC	87.77	95.82	82.06
	MCC	47.96	88.23	62.65
LSTM Autoencoders (Compression = 0.4)	ACC	89.08	92.33	77.30
	MCC	53.83	78.35	50.90
LSTM Autoencoders (Compression = 0.8)	ACC	88.43	67.83	77.90
	MCC	50.42	39.61	51.39

Table 3. Comparison of the performance of the hybrid metaheuristic over various state-of-the-art approaches.

Work	#Features	Approach	Dataset	Accuracy (%)
Toolan and Carthy [16]	22	Content and behavior-based features	Phishing: 6,458 Non-phishing: 4,202	Set *a*: 97.00 Set *b*: 84.00 Set *c*: 79.00
Zhang *et al.* [20]	7	Behavior-based features	Host-based: 2,328	Train: 95.80 Test: 99.60
Ma *et al.* [13]	7	Content and behavior-based features	Phishing: 46,525 (7%) Non-phishing: 613,048	99.00
Hamid and Abawajy [9]	7	Content and behavior-based features	Set *a* (H and P): 1,645 Set *b* (H and P): 2,495 Set *c* (H and P): 4,594	Set *a*: 96.00 Set *b*: 92.00 Set *c*: 92.00
Zareapoor and Seeja [19]	Varying ($10 - 2,000$)	Content and behavior-based features with dimensionality reduction	Phishing: 1,000 Non-phishing: 1,700	LSA[a]: 96.80 IG[b]: 94.20 PCA[c]: 96.40 χ^2: 94.50
This work	Set *a*: 164 Set *b*: 167 Set *c*: 172	Content and behavior-based features with dimensionality reduction	Set *a* (H and S): 3,051 Set *b* (H and P): 3,344 Set *c* (H, S, and P): 3,844	Set *a*: 99.78 Set *b*: 99.40 Set *c*: 98.79

[a] *Latent Semantic Analysis*, [b] *Information Gain*, [c] *Principal Component Analysis*; H: *Ham*, S: *Spam*, and P: *Phishing*.

4.2 Results and Discussion

The experiments were performed using a server running Ubuntu OS with two cores of Intel Xeon processors, 8 GB RAM, and one NVIDIA Tesla C-2050 GPU. All the algorithms were implemented in Python 2.7. To effectively test the performance of the proposed metaheuristic, we divide the entire email corpus using $70 - 30$ train-to-test split percentage. To facilitate exhaustive benchmarking of the proposed approach, we utilize Accuracy (ACC) and MCC scores as the evaluation metrics. Furthermore, we compare performance of the proposed Cuckoo-Firefly-GR over the base optimization approaches (Cuckoo-GR and Firefly-GR), dense autoencoders (two variants: four encoding layers, a compression factor of 0.4, and trained for 25 epochs and eight encoding layers, a compression factor of 0.8, trained for 50 epochs), and LSTM autoencoders (two variants: one encoding layer, a compression factor of 0.4, trained for 25 epochs and one encoding layer, a compression factor of 0.8, trained for 50 epochs). Table 2 compares the performance of various dimensionality reduction approaches. We observe that the feature subset obtained using hybrid metaheuristic, when classified using an MLP model outperforms various other feature selection approaches.

From the statistics of the email corpus and Table 1, it can be observed that the dataset is class imbalanced. MCC score facilitates a balanced score even in an imbalanced scenario by considering true and false positives and negatives [4]. From Table 2, it can be remarked that the feature subset obtained using the proposed Cuckoo-Firefly-GR results in comparatively higher MCC scores (closer to +1), indicating the superior predictive capability of the proposed approach. Furthermore, the better and fitter solutions obtained using the proposed metaheuristic can be attributed to the enhancement of balance between diversification and exploration, resulting in superior performance in accuracy.

Our results also signify the impact of capturing syntactic and semantic email content features. Table 3 compares the proposed approach with various state-of-the-art methods. Early works [5,1] used email content-based features to aid in the UBE classification. The results presented by Chandrasekaran *et al.* [5] show that a better classification is obtained with a larger number of features. Abu-Nimeh *et al.* [1] showed that an MLP model outperforms other machine learning classifiers. Several recent works [20,13,16,10] establish the need for both content-based and behavior-based features in phishing email classification. Zareapoor and Seeja [19] establish the need for mining email body content and using effective dimensionality reduction techniques to classify the email types effectively. This work employs a bio-inspired metaheuristic to mimic the best features in nature including natural selection, reproduction, and social behavior. The metaheuristic optimally selects a discriminative feature subset needed for the classifier to learn and generalize. Using an MLP model, we obtained an overall accuracy and MCC score of approximately 99% for all three UBE datasets. The large heterogeneity in the datasets used in various existing studies makes it challenging to compare their results with the proposed approach, despite which, our results are robust and comparable in terms of the overall accuracies and MCC scores.

5 Summary

UBE classification is a challenging problem due to the dynamic nature of UBE attacks. In this paper, we used forty prominent content and behavior-based features used in the existing literature in addition to the Doc2Vec features of email message content. Doc2Vec modeling captures the syntactic and semantic textual features in the email content. We also designed Cuckoo-Firefly-GR, a hybrid metaheuristic to obtain a discriminative and informative feature subset crucial to UBE classification. The proposed Cuckoo-Firefly-GR combines the concepts of random walks in CS with the mechanisms of FA such as attractiveness and intensity to enhance the optimal balance between diversification and exploration. Moreover, the hybrid metaheuristic incorporates the evolutionary strategies of GA including selection, crossover, and mutation, to genetically replace the lower fitness individuals resulting in faster convergence and superior performance.

The proposed algorithm has been extensively tested using a corpus of $3,844$ emails to evaluate its efficacy. We underlined the superiority in the performance of the proposed Cuckoo-Firefly-GR over various feature selection methods, including base optimizations (Cuckoo-GR and Firefly-GR), dense autoencoders, and LSTM autoencoders. We also presented a comparative analysis of the proposed method over various state-of-the-art methods. The discriminative feature subset obtained by the proposed hybrid metaheuristic, when classified using an MLP model resulted in the overall performance of 99%. Our results revealed the impact of using effective email content modeling strategies along with efficient feature selection approaches in the classification of email types. In the future, we aim at extending the proposed approach to model the graphical features popularly used to exploit content and behavior-based anti-UBE mechanisms.

References

1. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit. pp. 60–69. ACM (2007)
2. Arora, S., Singh, S.: A conceptual comparison of firefly algorithm, bat algorithm and cuckoo search. In: 2013 International Conference on Control, Computing, Communication and Materials (ICCCCM). pp. 1–4. IEEE (2013)
3. BİRİCİK, G., Diri, B., SÖNMEZ, A.C.: Abstract feature extraction for text classification. Turkish Journal of Electrical Engineering and Computer Sciences (2012)
4. Boughorbel, S., Jarray, F., El-Anbari, M.: Optimal classifier for imbalanced data using matthews correlation coefficient metric. PloS one **12**(6), e0177678 (2017)
5. Chandrasekaran, M., Narayanan, K., Upadhyaya, S.: Phishing email detection based on structural properties. In: NYS cyber security conference. pp. 2–8 (2006)
6. Elkhechafi, M., Hachimi, H., Elkettani, Y.: A new hybrid cuckoo search and firefly optimization. Monte Carlo Methods and Applications **24**(1), 71–77 (2018)
7. Gangavarapu, T., Jaidhar, C.D., Chanduka, B.: Applicability of machine learning in spam and phishing email filtering: review and approaches. Artificial Intelligence Review pp. 1–63 (2020)
8. Gangavarapu, T., Patil, N.: A novel filter-wrapper hybrid greedy ensemble approach optimized using the genetic algorithm to reduce the dimensionality of high-dimensional biomedical datasets. Applied Soft Computing **81**, 105538 (2019)
9. Hamid, I.R.A., Abawajy, J.: Hybrid feature selection for phishing email detection. In: International Conference on Algorithms and Architectures for Parallel Processing. pp. 266–275. Springer (2011)
10. Hamid, I.R.A., Abawajy, J.H.: An approach for profiling phishing activities. Computers & Security **45**, 27–41 (2014)
11. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196 (2014)
12. Lozano, M., García-Martínez, C.: Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. Computers & Operations Research **37**(3), 481–497 (2010)
13. Ma, L., Yearwood, J., Watters, P.: Establishing phishing provenance using orthographic features. In: eCrime Researchers Summit, 2009. eCRIME'09. IEEE (2009)
14. de Oliveira, V.Y., de Oliveira, R.M., Affonso, C.M.: Cuckoo search approach enhanced with genetic replacement of abandoned nests applied to optimal allocation of distributed generation units. IET Generation, Transmission & Distribution **12**(13), 3353–3362 (2018)
15. Symantec: Internet security threat report. Tech. rep. (March 2018)
16. Toolan, F., Carthy, J.: Feature selection for spam and phishing detection. In: 2010 eCrime Researchers Summit. pp. 1–12. IEEE (2010)
17. Yang, X.S.: Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. pp. 169–178. Springer (2009)
18. Yang, X.S., Deb, S.: Cuckoo search via lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). pp. 210–214. IEEE (2009)
19. Zareapoor, M., Seeja, K.: Feature extraction or feature selection for text classification: A case study on phishing email detection. International Journal of Information Engineering and Electronic Business **7**(2), 60 (2015)
20. Zhang, J., Du, Z.H., Liu, W.: A behavior-based detection approach to mass-mailing host. In: 2007 International Conference on Machine Learning and Cybernetics. vol. 4, pp. 2140–2144. IEEE (2007)