Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.086257

ARTICLE

A Novel Metaheuristic Approach for Phishing Websites Detection with the Modified Differential Evolution Algorithm

Mohammad Alshinwan[1,*], **Walaa Alayed**[2,*], **Fatma A. Hashim**[3] **and Arar Al Tawil**[4]

[1]Department of Computer Science, Faculty of Information Technology, Al al-Bayt University, Mafraq, Jordan
[2]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia
[3]Biomedical Engineering Department, Faculty of Engineering, Capital University (Formerly Helwan University), Cairo, Egypt
[4]Computer Sciences Department, Faculty of Information Technology, Applied Science Private University, Amman, Jordan

*Corresponding Authors: Mohammad Alshinwan. Email: shinwan@aabu.edu.jo; Walaa Alayed. Email: wmalayed@pnu.edu.sa

Received: 27 May 2026; Accepted: 06 August 2026; Published: 15 September 2026

ABSTRACT: The increasing trend of phishing sites is among the important threats against the Internet security, associated with monetary loss, data leakage, and identity swindle. In response to this urgent problem, this paper proposes a new phishing website detection framework based on the Modified Differential Evolution (mDE) algorithm in conjunction with state-of-the-art machine learning classifiers. The proposed mDE integrates with dynamic mutation and crossover strategies to improve the global search capability and the convergence speed, which is superior to traditional single optimization methods. We conduct experiments on two benchmark datasets: the UCI Phishing Websites dataset and the large-scale Mendeley Phishing URLs dataset. The proposed optimizer is implemented as an improved variant, mDE+, which augments the adaptive control parameters with opposition-based initialization, guided mutation, and self-adaptive control of the mutation and crossover factors. Using Random Forest and gradient-boosting classifiers, mDE+ selects compact feature subsets that are competitive with, and on most settings superior to, those produced by standard DE and other nature-inspired optimizers (GWO and COA), with statistically significant gains over DE and COA on UCI and negligible additional computational overhead. Experimental results demonstrate that the proposed approach outperforms the state-of-art methods in terms of accuracy, precision, recall and F1-score, which further verify the effectiveness and robustness of the proposed method. This paper shows that mDE is a promising optimization method for training phishing detection systems for real-world cyber security applications.

KEYWORDS: Websites phishing; meta-heuristics optimization algorithms; differential evolution; random forest; gradient boosting

1 Introduction

The rise in the implementation and the use of information technology (IT) has remarkably expanded the prevalence of web services, which are available in different domains ranging from financial transactions, learning platforms to e-health services. Recent statistics indicate that cyber cash transactions, social networking platforms, and gaming services in particular have become increasingly popular, attracting millions of users across the world. The pervasive and habitual use of these online services highlights the critical nature of online service availability and reliability that are required for daily life. However, this level of reliance and constant availability of computer systems also inadvertently increased the attack surface, rendering

computer-based systems to a greater variety of cyber threats [1]. Amid these threats, the phishing attack on websites has arisen as one of the most popular and devastating examples.

The increasing sinuosity and growth of phishing websites signal a steady rise in the intensity of the threat in the current cyber world, which could eventually result in the leakage of personal or enterprise information, loss of systems, and disruption of global cyber peace. There has been a notable surge in the volume and sophistication of phishing attacks, where attackers use various tricks to steal sensitive information, including usernames, passwords, financial information, and personal details, making it one of the most challenging security threats to tackle. Recent statistics reveal the alarming state of affairs, indicating a rapid rise in successful phishing attacks and substantial financial losses worldwide.

The development of phishing websites has been escalating shockingly in large scale, chain crime and economic loss. The Anti-Phishing Working Group (APWG) observed close to 5 million phishing attacks in 2023, with over a million in the final quarter of the year alone [2]. Vicious phishing domains, however, mostly have a lifetime of about 11.5 days, which gives attackers enough time to quickly take advantage [3]. According to IBM, phishing is the second most expensive attack vector with an average loss in the range of USD 4.76 to 4.88 million per breach, with global averages measuring to nearly USD 4.9 million, an increase of over 10% from previous years [4]. Moreover, adversaries increasingly weaponize artificial intelligence to orchestrate tailored spear-phishing campaigns, producing dynamic lures that routinely bypass static, rule-based defenses [5]. Today, phishers have upped their game by social media, webmail, SaaS platforms, and other new techniques, such as SMS "smishing" or voice "vishing" to expand their attack surface [6,7]. Therefore, phishing remains to be a severe and emerging cybersecurity threat. Since phishing website threats are becoming more severe and have significant impacts, the effective and accurate detection methods are necessary. Machine learning (ML) has attracted attention as a powerful approach to phishing website detection by examining various features (e.g., structural factors, webpage content, and user activity). Yet, the performance of ML methods critically depends on hyperparameter tuning, which in turn affects the accuracy and generalization ability of prediction models. However, traditional hyperparameter optimization approaches such as expert knowledge and grid search are usually time and computation consuming and not always leading to the best performance.

Recent developments in the field of phishing website detection have focused in improving nature-inspired metaheuristic optimization such as improvement of nature inspired metaheuristic optimization and improving accuracy of machine learning classifiers. For example, swarm intelligence algorithms, such as Bat Algorithm [8], Firefly Algorithm [9], Grey Wolf Optimizer [10], and Whale Optimization Algorithm [11], have been employed to improve the capability of distinguishing the phishing sites using URL features for support vector machines (SVM). Results illustrate that the GWO performed better than the other methods in finding the optimal hyperplane for the classification problems [12]. Besides, the particle swarm optimization (PSO) has been successfully used by the researchers to weigh the site characteristics, which have greatly enhanced phishing detection score with lower false positive and false negative [13]. Also, the Arithmetic Optimization Algorithm (AOA) was found to be more accurate and efficient than PSO, Multi-Verse Optimizer, and Salp Swarm Algorithm for selection of relevant features for effective classification of malicious URLs, based on classification accuracy [14]. Moreover, combining SVM with ACO, along with a DBN, yielded strong feature selection among large sets of phishing and legitimate URLs, leading to a substantial improvement in the accuracy of detection [15]. Finally, new methods such as the k-Nearest Neighbors (k-NN) models combined with the Harris Hawks Optimizer (HHO) have obtained interesting results to address high-dimensional data, outperforming the spam detection results of Binary Dragonfly Algorithm (BDA), Equilibrium Optimizer (EO), Teaching-Learning based Optimization (TLBO), Seagull Optimization Algorithm (SOA) and Marine Predators Algorithm (MPA) as other optimization techniques.

To overcome the structural shortcomings of standard Differential Evolution in phishing detection pipelines, we propose a modified Differential Evolution (mDE) variant featuring adaptive control parameters and dynamic search strategies to stabilize convergence and preserve population diversity. Building on this core formulation, we implement an extended variant, mDE+, which pairs opposition-based initialization with a current-to-pbest mutation operator and JADE-style parameter self-adaptation. When applied to tune Random Forest and gradient boosting models across benchmark phishing datasets, mDE+ consistently outpaces standard DE and related metaheuristics across key classification metrics, including accuracy, precision, recall, and F1-score.

Although Differential Evolution (DE) and its adaptive techniques have been shown to effectively optimize machine learning applications, their deployment towards a particular cybersecurity application (such as phishing website detection) is limited by a few issues. Conventional DE algorithms often reach premature convergence, limit the diversity of the population during later iterations, and imbalance between the exploration and exploitation processes. These limitations can hinder optimization efficiency when dealing with heterogeneous phishing datasets that exhibit intricate feature interactions. Adaptive Generalized Differential Evolution (AGDE) provides greater parameter adaptation and convergence behavior, but still relies heavily on population differences and lacks an explicit transition mechanism to regulate exploration and exploitation throughout the optimization process. Moreover, search diversification is still limited under highly nonlinear feature spaces. To overcome them, this work introduces a modified Differential Evolution algorithm (mDE) designed to improve optimization stability, population diversity, and global search capability. The proposed approach makes several modifications that set it apart from conventional DE variants:

- Bidirectional randomization factor (RF): The introduced RF, unlike traditional random parameters being restricted to positive search directions, produces both positive and negative perturbations, further diversifying the population and decreasing premature convergence.
- Adaptive transition factor (α): A transition mechanism is added to gradually move the optimization from global exploration to local exploitation, which increases convergence stability over iterations.
- Modified mutation strategy: Rewrite the mutation operator to better focus exploration without losing local optimization near promising parts.
- Oscillatory exploration mechanism (β): The adaptive exploration parameter is added to lead to a diversified search behavior during the early steps of the search and to improve convergence timewise in the latter stages.
- Application-specific optimization for phishing detection: Whereas traditional DE studies mostly measure the performance of classifiers on benchmark functions, we validate the proposed mDE+ with multiple phishing datasets to optimize machine learning classifiers against simulated real cybersecurity scenarios.

Thus, these modifications aim to enhance the optimizations robustness and classification performance in phishing detection with dynamic and high-dimensional feature spaces.

This paper propose the following:

- We formulate a Modified Differential Evolution (mDE) framework tailored for feature optimization in phishing website detection pipelines.
- The core mDE architecture introduces four coordinated mechanisms—a bidirectional Randomization Factor (RF), an adaptive Transition Factor (α), a revised mutation operator, and an oscillatory parameter (β)—to counter diversity loss and stabilize exploration-exploitation dynamics.

- We develop an extended variant, mDE+, which integrates opposition-based population initialization, a current-to-pbest mutation scheme, and JADE-inspired adaptive tuning for mutation scale (F) and crossover rate (CR).
- We couple the proposed optimizers with Random Forest and Gradient Boosting ensembles to construct high-accuracy classification pipelines.
- Extensive benchmarking across standard phishing repositories demonstrates that mDE+ consistently outpaces classical DE and competing nature-inspired baselines, delivering statistically significant gains over standard DE and COA on the UCI benchmark.

The rest of the paper is organized as follows: Section 2 reviews the recent study related to the detection of phishing websites. Section 3 presents the proposed mDE algorithm. Section 4 shows the methodology of the experiments and the results obtained. Section 5 discusses the results achieved. Finally, Section 6 presents the conclusion and potential future works

2 Related Studies

Phishing has become one of the most prevalent cyber threats, targeting individuals and organizations alike by impersonating trusted entities to steal sensitive information. Detecting phishing sites effectively remains challenging due to the rapid evolution of phishing tactics. As phishing schemes grow more complex, researchers have developed diverse detection methods to counteract these attacks. Techniques range from list-based approaches that rely on blacklists and whitelists to more advanced machine learning and deep learning models that analyze website characteristics. This section reviewes the main approaches used for phishing website detection, with particular emphasis on heuristic techniques and machine learning methods, highlighting each method's strengths, limitations, and recent advancements.

2.1 List-Based Techniques in Phishing Detection

Early phishing detection methods built on a lists-based approach are probably the most widely adopted method to detect fishy websites. List-based detection comprises the use of two types of lists: blacklists consistof website addresses determined to be phishing sites, and whitelists which are comprise of website addresses that have been verified as being safe. Commonly used by browsers such as Google Chrome, Firefox, and Edge, this approach offers a straightforward mechanism: a site listed on the blacklist is blocked, while access to whitelisted sites is allowed with no further checks. With the help of security experts or users, PhishTank and Google Safe Browsing sites have databases that contain a number of these banned URLs. However, while effective in blocking documented phishing sites, blacklists face significant challenges with "zero-hour" phishing sites, or newly created URLs that have not yet been reported. Taking advantage of the fact that new phishings URLs will not be on the blacklist, perpetrators create phishing URLs specifically to be undetectable, making this one of the key limitations of this approach of list-based detection systems [16]. The strengths of list techniques come from their ease of use and low computational requirements. For instance, Ref. [17] showed that blacklists are very effective in curbing phishing threats especially when they are integrated with a browsers warning system that cautions the users on certain sites. However, such simplicity has its own drawbacks: the restriction to low level phishing attacks means that constant updates are required so that the consumer can be appropriately protected against a new threat. Ref. [18] pointed out, all it might take for a would-be attacker to bypass such protection mechanisms is a single URL that has not been added to the blacklist, thus putting the user's data at risk despite the many efforts being made to create and regularly update these blacklists. Also, as the length of such lists increase, the cost of frequently updating them and the needed amount of space for storage may prove a challenge particularly for firms with

fewer resources. To address these problems, new combined techniques have been developed, integrating list-based techniques with heuristic or machine learning methods. For instance, Ref. [19] reported that they have employed blacklist filtering and the Random Forest classifier that managed to obtain 97 percent accuracy. This hybrid system allowed the prediction of phishing patterns in URLs that are in the blacklist. It enhanced the model's ability to adapt to the evolution of new phishing techniques. Likewise, Ref. [20] achieved the same high accuracy of 99.33% when they integrated list-based data with the PART rule-based classifier together with the data. Their strategy also resolved the problem of "lookalike" phishing sites that have nearly the same URLs but are not on the blacklist by employing rules to search for sites. As phishing attempts become more sophisticated, list-based procedures also evolve. As mentioned in the current research, systems are being developed that would enable real-time modifications, as well as proactive flagging of suspicious URLs. Rao and Pais [16] reviewed several existing anti-phishing techniques, including blacklist- and whitelist-based approaches, and introduced a feature-based machine learning framework to improve phishing website detection. Their method uses heuristic features extracted from URLs, webpage source code, and third-party services, which are then evaluated using eight different machine learning algorithms. The results showed that Random Forest achieved an accuracy of 99.31%, while combining PCA with Random Forest increased the accuracy to 99.55%. The framework also showed promising performance in detecting zero-day phishing attacks, highlighting the ability of machine learning methods to identify malicious websites that may not yet appear in traditional blacklists. To conclude, while list-based techniques are an essential part for strengthening the anti-phishing barrier, their utility is not without limitations. For example, while these methods are effective in retrospect, their primary drawback is that they rely heavily upon continuous maintenance of databases and there is a lack of functionality to cater for new phishing URLs which indicates that there is a need for improvement. The scope for caveats can be established in future evolutions due to including machine-learning algorithms and predictive analytics alongside list-based filtering, which will make such systems more fluid in time.

2.2 Visual Similarity-Based Techniques in Phishing Detection

In the case of visual similarity-based approaches, according to the authors, it enables a more advanced detection of phishing attacks by the aid of a webpage's features and enables the detection of look-alike phish websites by comparing logos, layout and designs. The most common methods which are a list-based methods are not suitable for this approach. Instead, it considers the page's structure and style characteristics and, therefore, is able to find lookalike sites of legitimate business. But recent trends show that these methodologies have become more important for users as this is rampant as attackers today create phishing pages which look identical to the real ones in order to get users sensitive information. Among the methods of visual similarity detection, one of the predominant methods is measuring different elements of the page such as Cascading Style Sheets (CSS), text and how the logos are positioned. Ref. [21] further evaluated this method by looking at visual markers that can identify a phishing site from legitimate one and in this case focusing on layout patterns and logos, they reached an accuracy of 98.05%. However, the application of this approach is limited especially for zero-hour phishing where more information is required to make a judgement on unfamiliar sites. Other studies have improved visual similarity assessments by adding techniques like Optical Character Recognition (OCR) and Fuzzy Logic in order to increase the precision. Ref. [22] applied a Fuzzy Soft Set (FSS)-based method to quantify visual similarities to lessen phishing attacks achieving a remarkable 99.77% accuracy rate. Concentrating on image-based components and structural similarities demonstrated great effectiveness in both the volume and range of phishing websites. Moreover, Ref. [23] proposed a model of logo recognition that is OCR-based and mitigated some shortcomings of text-based visual matching that has proven the working ability of OCR aids in brand name and layout similarity replication. Even though

they have advantages, the comparison of pictorial attributes does not have an easily available computational capacity, which is mostly required when such processes have to be performed in parallel, also assessed detection methods, which compare embedded images with images stored in a database of templates with a slow performance, since such methods are image recognition and searching based. One of the solutions was provided by [24] who estimated 99.66% accuracy by visual effects combined with filters relying on web crawlers. Consistency of detection results was, however, troubling being a function of the time the search engines had active results. In an attempt to overcome the dependence on static visual templates in visual detection methods, there is an urgent call, as highlighted by the researchers, to consider hybrid models of these techniques with machine learning. Pursuing this approach, future systems may decouple and revise visual models to reflect the evolution of the template systems' designs in an effort to reinforce the techniques of phishing detection. For instance, such vision technologies as neural networks, due to their recent success particularly in convolutional neural networks (CNNs), may have benefits of application in situations where there is deep level of style weakness that cannot be captured in previous visual recognition methods. Expanding these technologies is likely to enhance the complementarities of visual interrogation, making it possible for the system to cope with the rapidly changing strategies of the phishers.

In conclusion, visually deceptive phishing sites continue to evade standard defenses because they replicate target brands almost pixel for pixel. While similarity-based detection offers a practical safeguard, it carries real operational trade-offs—chiefly the overhead of rendering pages in real time and the constant need to refresh reference image libraries. Shifting from static screenshot matching toward adaptive, machine learning–driven baselines will be crucial for these detectors to keep pace as attackers automate and refine their evasion tactics.

2.3 Heuristic Techniques

Phishing detection heuristic techniques are unique since they can locate phishing sites by scanning the website for distinctive attributes. They do not depend on a list of URLs but heuristics try to target signs of phishing by scanning a web page. This approach employs uncommon proxies such as URL composition, domain name system (DNS) information, Secure Sockets Layer (SSL) certificates and other web materials to mark websites as suspicious. Heuristic methods are particularly effective in dealing with the zero hour type of phishing, whereby new phishing sites exist with no previous records or any known history. The major strength of heuristic approaches is the ability to locate sites through identifying patterns or anomalies. Ref. [25] trained a robust set of heuristic features using a Random Forest classifier, achieving 99.57% accurate results. In practice, the classifier evaluates structural heuristics—including registration age, SSL certificate validity, and string length—to flag anomalous domains before users land on them. However, while heuristic methods are able to provide reliable results they can also lead to false alerts for example when phishing sites have similar features to legitimate websites. For instance, a phishing site operating through a URL that is long or is not familiar might be marked as fishy because that URL is considered unusual. Some researchers argue that with more sophisticated heuristics that include machine learning, the accuracy is increased, Ref. [26] created a hybrid heuristic model that combines feature selection and nonlinear regression, and this model was successful in classifying phishing websites from legitimate ones. That model was based on URL features and web usage patterns, and it was able to achieve a high level of accuracy even for complex and intricate URL structures. The hybridization of heuristic techniques and ML methods provides more sense to the analysis and increases the performance of the system with regards to the adoption of new phishing strategies but not at the expense of the speed of detection. However, in addition to the analysis of the URLs and content of the site, heuristic methods frequently evaluate the security measures of the site. The use of SSL certificates, for instance, is a feature assessed in the heuristic models as phishing sites tend to have weak encryption or no

valid certificates at all. To deal with these challenges, Ref. [16] boosted the heuristic methods by adding the analysis SSL certificates and this increased the chances of finding phishing sites which did not follow standard security practices. Since on the one hand site characteristics that are visually perceivable and the other hand characteristics that are not visibly perceivable are addresses specifically by heuristic techniques a relatively balanced algorithm in view of the deceptive nature of phishing attacks is provided. Feature-based methods do have some inherent problems to them. When this reasoning is set forward, heuristic techniques are based on some specific rules. Thus, attackers can construct a site that aims to bypass all heuristics, for example, creating an HTTPS page or a domain that seems legitimate. To combat this problem, some recent heuristic methods work by integrating the techniques with models that are able to predict and evolve with time. Ref. [19] has proposed what they refer to as a heuristic technique that uses predictive analytics to improve its effectiveness over time. By predicting the new domain as if they had learned from past phishing incidents, the model could reduce the probability of a new phishing site being established. To conclude, heuristic techniques present an effective measure for the detection of phishing and other Internet-related scams. While the use of heuristics ensures that threats are quickly detected, there is a need to fortify these techniques with Machine Learning and predictive techniques in order to deal with more complicated phishing threats. In the future, it is hoped that research will have reached at a stage where systems even of high sophistication will have a stronger and more accurate heuristic method of detecting modern phishing techniques.

2.4 Machine Learning Techniques in Phishing Detection

The use of machine learning (ML) approaches in phishing detection has been a great game changer in the sense that such models are flexible, precise and can withstand more advanced and myriad phishing threats. This is in contrast to the static list-based systems or rule based systems, which are not ML driven. Learners are documented to have some level of indescribable efficacy in the suppression of patterns by grouping phishing sites based on a variety of characteristics from legitimate sites. Extracting or excerping features e.g., URL, web page meta information, Java Script and other site factors are commonplace in most machine learning approaches which train algorithms that help identify phishing or safe websites. As a consequence of its accuracy and robustness in performance, Random Forest has emerged as a popular machine learning algorithm used in phishing detection systems. Ref. [27] was able to attain a detection rate of around 99% after training a Random Forest model on a huge phishing site and legitimate site dataset. Some of the features, which include URL length, the presence of HTTPS, and other domain name attributes were extensively analyzed by the model. By uncovering non-linear relationships and subtle behavioral cues, the model reliably flags sophisticated phishing domains that evade heuristic and signature-based filters. Consistently high and reliable accuracy rates irrespective of the dataset used has been another key benefit of Random Forest in the detection of phishing sites as according to [28] eingesch it utilities a variety of web features to identify phishing sites. Additionally, Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) along with other machine learning models are also common in the detection of phishing activities. In their investigation, Ref. [29] managed to build optimised URL to SVM models which they reported to have a reasonable detection rate accuracy. The work of these authors involved taking constituent parts from the structures of URLs such as the numbers of subdomains and special characters and applying them to the SVM to categorize phishing URLs within very large datasets such as PhishTank. Moreover, Ref. [19] also reported that KNN could be used with other heuristic features, thus demonstrating how useful it is in detecting even subtle phishing differences between false and authentic URLs based on their feature space similarities. To increase accuracy and minimize false rates, machine learning techniques towards phishing detection can also be improved using ensemble techniques. According to [30], the Random Forest Support Vector Machines ensemble model they developed yielded an accuracy of 99.33% by selecting features using

Particle Swarm Optimization (PSO) and tuning hyperparameters using a Tree-structured Parzen Estimator (TPE). Such integration enabled the construction of a tightly configured model capable of recognizing the slightest phishing variations while retaining a very low level of false positives. Thanks to their effectiveness in coping with various techniques used in phishing attacks, and being able to deal with emerging techniques, Random Forests and other classifiers-based ensemble models are still the most widely used models in phishing prediction.

The ability of these competency models to work with persistent significant volumes of data sets available is key given the amount of potential phishing attacks on a daily basis. For example, studies such as those by [27] emphasize the scalability of machine learning since it enables the fast processing of URL classification in lots of sites at once, thus it has capacity for adoption in huge security networks. Moreover, natural language processing (NLP) has aided phishing detection by enabling the analysis of textual components such as URLs, emails, and webpage content, while computer vision addresses the visual appearance of web pages; these are distinct sub-fields of machine learning. Through NLP, models can infer subtle linguistic cues that may indicate phishing [31], which is a further advantage of ML-based systems. Nevertheless, machine learning has its own pitfalls. First of all, to be precise and accurate and to a large extent train these models, it still requires an enormous amount of data which in this case are the URLs of phishing episodes vs. those that are legitimate. Moreover, it is known that algorithms conditioned on large data bases are good in recognition, yet such models might be at a loss during a phishing attack that utilizes modern cognitive infiltration techniques or graphical structure containing images or videos. The researchers are still making efforts to address these limitations as they attempt to combine artificial intelligence and deep learning which continues to develop to train the models and defend against more complicated phishing campaigns while still performing well against other traditional techniques.

Thus machine learning techniques have been found to be quite helpful in combating phishing attacks and they are probably the best defense in terms of accuracy, flexibility as well as being cost effective. Building upon the previous sentence, as these techniques and approaches continue to progress, it is reasonable to assume that combining them with other methodologies, such as deep learning and heuristic methods, will result in even more powerful detection systems that will be able to combat the dynamic and changing nature of phishing approaches.

2.5 Deep Learning Techniques in Phishing Detection

Phishing detection has benefited greatly from the innovations imparted by deep learning (DL) methods, which allow for the use of highly complex and multi-layered neural networks that can identify intricate designs and relationships in high dimensional space data. Unlike conventional machine learning models which depend heavily on hand-crafted features, DL's approach of automatic detection of features as well as learning to be able to identify even the tiniest details of indications of phishing, allows deep learning models to be appropriate for fighting phishing that keeps changing with its sophisticated nature. Today's security experts, for example, have observed a rise in the use of models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for phishing detection which has been made possible due to completeness handled by these models such as text, images as well as different time series.

The deep learning paradigm offers its users the possibility of mining large datasets that are complex in nature while uncovering underlying relationships that can go unnoticed by simplistic models. This paper [32] constructed a CNN-based model that triggered phishing websites after analyzing character levels of the URL which enabled the model to detect zero-day phishing websites. That is, by analyzing the character-level composition of each URL, the model was trained to distinguish previously unseen phishing URLs from legitimate ones. This kind of analysis on the character level gives CNNs the ability to detect domain URLs

that are fake website replicas, and this makes deep learning models efficient in preventing fraudulent domain names from use.

Other than CNNs, RNNs and LSTMs also find a considerable amount of application in the field of phishing detection, especially on tasks that involve sequential data. Ref. [31] proposed an RNN-based phishing detection model that processes the sequential character and token structure of URLs and page content. By retaining information across the input sequence, the model captures dependencies among successive elements of a URL, which improves the detection of phishing patterns. So, RNNs and LSTMs are also well suited for phishing detection when these sequential cues are available, like in emails and web pages navigation history.

Researchers have proposed hybrid approaches that add natural language processing, and image analysis to strengthen phishing detection based on deep learning. Ref. [31] proposed the Web2Vec model which added multi-dimensional feature analysis with deep learning to improve of harvesting of phishing pages. This model used natural language processing to study the language of the enterprise websites and deep learning to search for trademark phishing across the mediums making it possible for phishing attempts to be detected with high sensitivity and accuracy. Web2Vec illustrated the potential of deep learning when combined with natural language processing and convolutional neural networks which were effective in handling the complex nature of phishing attacks, thereby preventing attacks that were likely to bypass mainstream filters.

Deep learning models further extend their usefulness in image-based phishing detection where phishers aim to copy the look of genuine websites. In this context, it is very useful to have Convolutional Neural Networks which are optimized for visuals recognition. Screenshots or layouts can be examined by the CNN and any deviations in the location of logos, colors, and visual designs which are usually present in phishing attempts will be detected. Ref. [23] in one of their works used a CNN model on thousands of images of real-life and phishing websites and the model achieved high levels of performance in identifying features present in phishing websites. Thus, this makes it possible for the CNNs to detect sites that are built for phishing and visually impersonate genuine ones. Nonetheless, deep learning models face some hurdles, including the need for sufficiently large datasets and immense amounts of processing power. Even though, for instance, CNNs and RNNs are highly accurate, their high complexity often results in long turnaround times that are not ideal for real-time detection and for environments where resources are limited. Also, some deep learning models have the problem of overfitting, which occurs when a model is so tailored to the training set that it does not work well with other data sets. Some researchers, however, are trying to solve these problems by adapting transfer learning where instead of training a model from scratch, a model pre-trained on general datasets is fine-tuned on a domain-specific phishing dataset to reduce the amount of data and time needed for training while maintaining accuracy. To sum up, in addition to enhancing the detection accuracy in conjunction with traditional methods, deep learning technologies are already a potent weapon against phishing site detection for their ability to abstract and structure complex, high-dimensional data points that cannot be achieved by conventional techniques. With better resources, deep learning models are however difficult to ignore on account of their versatility in detecting finer details that other models would easily overlook which are fundamental in countering the much more advanced phishing attacks. In the future, combining deep learning techniques with other approaches, such as visual approaches and language analysis could be beneficial in improving the performance of anti-phishing systems as more research in the subject gives way to new findings.

Conventional phishing detection strategies have gained a considerable amount of ground in the past few years, however, every strategy has its own advantages and disadvantages. List-based strategies are easy to implement and are helpful for prospective attacks, however, old phishing sites that are not present in the list are an issue. Visual similarity-based methods along with heuristic techniques are more efficient since

they assess web pages, but high processing power is required. On the other hand, machine learning and deep learning models are more effective and accurate as they have many algorithms that can prevent even slight phishing attacks. They are usually trained with many examples, thus costly computational resources are necessary. Constructing a fusion system in the future may be the best way to go as this will allow the systems to be effective against the different forms of phishing attacks that are likely to arise. As a start, using a combination of machine learning, visual analysis and heuristic models their possible approaches would be a great improvement on shielding them from potential attackers.

2.6 Research Gap

The reviewed literature shows that metaheuristic feature selection and hyperparameter optimization can improve phishing detection, and recent studies continue to advance differential evolution and its variants for high-dimensional feature selection [33–35]. Nevertheless, three gaps persist. First, most DE-based phishing studies rely on a single dataset and do not report the intrinsic computational cost of the optimizer or its overhead relative to standard DE. Second, the reproducibility of the reported gains is rarely established through released code and explicit hyperparameter settings. Third, newer and more challenging phishing corpora (e.g., PhiUSIIL [36]) expose the limited generalization of models tuned on older benchmarks. The present work targets the first two gaps directly: it introduces an improved and fully reproducible DE variant (mDE+), reports a complete cost analysis, and evaluates the method transparently on two public benchmarks, and validating the method on the newest corpora, namely PhiUSIIL [36] and StealthPhisher.

3 Proposed Methodology

This section outlines the end-to-end framework for detecting phishing websites via metaheuristically optimized machine learning classifiers. As illustrated in the workflow, the detection pipeline spans data ingestion, targeted preprocessing, feature selection and hyperparameter tuning driven by the proposed mDE+ optimizer, and rigorous multi-metric validation.

This study uses the feature representations provided by the two public benchmark datasets rather than a custom crawling pipeline, which ensures reproducibility and avoids extraction-time bias. The UCI dataset supplies 30 pre-computed categorical indicators (each encoded in $\{-1, 0, 1\}$) covering address-bar, abnormal, HTML/JavaScript, and domain-based properties, as defined by its providers [37]. The Mendeley dataset supplies 41 numerical lexical and host-based descriptors (e.g., URL length, counts of special characters, sub-domain statistics, and URL/domain entropy) as defined by its authors [38]. As both datasets are released fully pre-extracted with no missing values, no additional extraction, API look-ups, or failure handling are required. Numerical attributes were standardized, and, where needed, class balance was addressed by stratified sampling. This keeps the preprocessing consistent across optimizers and classifiers prior to feature selection and training. This provides a concrete conceptual representation of our proposed approach, where the procedural architecture of the mDE enhanced phishing website detection system at a high level is shown in Fig. 1. The organizational pattern is composed by three different phases of operation: data pipeline, the main modified Differential Evolution loop, and classifier evaluation block.

During Phase I (Data Pipeline phase, heterogeneous website instances from our target datasets are ingested and acted as a sequence of steps sequentially through data blocks. This phase performs MinMax scaling for numerical normalization, and class balancing operations with random under-sampling for balanced class distribution of the model(s) through random under-sampling for training equilibrium through this approach.

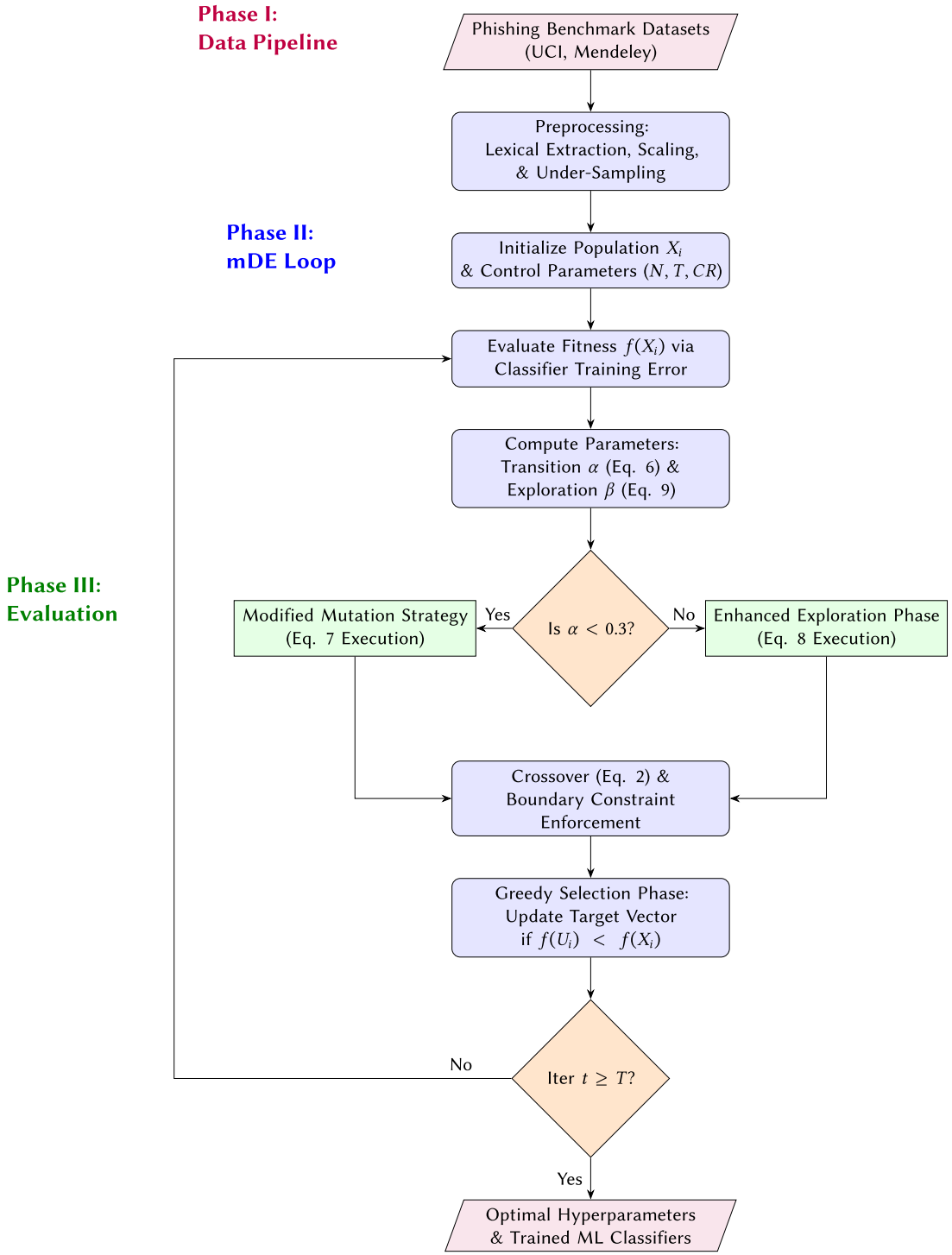

Figure 1: Overall logical flowchart of the proposed mDE optimization framework and phishing website detection pipeline.

Phase II is the main metaheuristic novelty of this article's contribution. Instead of static optimization parameters, the framework dynamically adjusts search vectors at each iteration. Transition Factor (α) governs the choice between exploratory global steps and fine-grained local refinement. At $\alpha \geq 0.3$, the model executes the Enhanced Exploration Phase, optimizing an oscillatory exploration parameter (β) for discovery of

regions on the high dimensional feature space that are not examined and avoiding local optima stalemate. On the other hand, when α becomes under 0.3, the technique easily switches over to local exploitation by the Modified Mutation Strategy. This adaptive decision is succeeded by the classic crossover recombination, boundary constraint validation, and phase of greedy selection that keeps elite candidate solutions.

Finally, in Phase III (Evaluation), the best hyperparameter arrays and feature sets produced at the convergence for further fine-tuning of downstream machine-learning models, namely Random Forest and gradient boosting, are processed. We validate the fine-tuned ensembles across a 10-fold cross-validation scheme to curb variance, confirm generalizability, and test whether their inference stability holds up under production-grade network constraints.

3.1 Review of the Original AGDE Algorithm

The Adaptive Generalized Differential Evolution (AGDE) [39] algorithm is an extension of the classical Differential Evolution (DE), designed to improve convergence performance by dynamically adjusting control parameters and promoting population diversity. AGDE preserves the core evolutionary mechanisms of DE—mutation, crossover, and selection—but augments them through adaptive strategies and multiple mutation schemes.

Initialization.

AGDE begins by initializing a population of N candidate solutions $X_i = [x_{i1}, x_{i2}, ..., x_{iD}]$, where D is the problem dimension. Each vector is randomly initialized within the problem bounds.

Mutation Operator.

AGDE incorporates several mutation strategies, one of the most prominent being:

$$V_i = X_r + F \cdot (X_a - X_b) \tag{1}$$

where X_r, X_a, X_b are randomly selected individuals from the population, and $F \in [0, 2]$ is the mutation factor. This strategy is referred to as "DE/rand/1". To promote adaptability, AGDE selects among multiple mutation strategies based on their historical success rates.

Crossover Operator.

The trial vector $U_i = [u_{i1}, u_{i2}, \ldots, u_{iD}]$ is generated by combining the mutant vector V_i and the target vector X_i:

$$u_{ij} = \begin{cases} v_{ij}, & \text{if } rand_j \le CR \text{ or } j = j_{rand} \\ x_{ij}, & \text{otherwise} \end{cases} \tag{2}$$

where $CR \in [0, 1]$ is the crossover rate, $rand_j$ is a random number for each dimension, and j_{rand} is a randomly chosen index to ensure at least one dimension is inherited from the mutant vector.

Selection.

AGDE applies greedy selection between the trial vector U_i and the original individual X_i:

$$X_i^{(t+1)} = \begin{cases} U_i, & \text{if } f(U_i) < f(X_i) \\ X_i, & \text{otherwise} \end{cases} \tag{3}$$

where $f(\cdot)$ denotes the objective function to be minimized.

Adaptive Parameter Control.

Unlike traditional DE, AGDE introduces adaptation mechanisms for F and CR using historical feedback from previous iterations. For example, a successful mutation or crossover leads to reinforcement of the associated parameters, using schemes such as:

$$F_{new} = F_{mean} + \tau_1 \cdot (F_{best} - F_{mean}) \tag{4}$$

and similarly for CR. Here, τ_1 is a learning coefficient.

Limitations.

Despite its improvements, AGDE can still suffer from limited search directionality and rigid exploitation behavior. The original strategy relies heavily on population differences without incorporating multi-directional or randomized shifts. These limitations motivated the proposed enhancements detailed in Section 4.2.

3.2 Proposed mDE Algorithm and Enhanced mDE+ Implementaion

This section details the proposed mDE algorithm and its enhancements over the original AGDE, particularly in improving search capability and balancing exploration and exploitation phases. The modified mDE incorporates four distinct components:

- Randomization Phase
- Transition Factor
- Modified Mutation Strategy
- Enhanced Exploration Phase

Randomization Phase

The randomization parameter is a key control element in any metaheuristic algorithm. In the original AGDE, a random value within the range $[0,1]$ is used. To improve the flexibility of the search, we introduce a new control parameter RF that generates both negative and positive values:

$$RF = 2 \times rand - 1 \tag{5}$$

where rand is a uniformly distributed random number within the interval $[0,1]$, yielding RF values ranging from $[-1,1]$. The proposed bidirectional randomization mechanism allows for both positive and negative directional movements (i.e., across the search space) rather than simply the positive perturbation-centric random parameters employed in traditional algorithms. From an optimization standpoint, the addition of bidirectional perturbation makes the population richer, and prompts broader exploration of candidate solutions in early iterations. The greater diversity prevents individuals from clustering around local optima too early in the search process; thus reducing the probability of premature convergence. As a result, RF improves global search capability and the algorithm's potential to transcend less optimal regions, resulting in more robust optimization performance for high-dimensional phishing detection tasks.

Transition Factor (α).

A major drawback of the AGDE algorithm is the absence of a transition mechanism between the exploration and exploitation phases. This limitation often results in unstable search behavior and increased computational time. To address this issue, a new parameter called the Transition Factor (α) is introduced. A common transition mechanism uses an exponentially decaying parameter, defined as:

$$\alpha = exp - \frac{t}{T} \tag{6}$$

The Transition Factor (α) is the balance that determines between the exploration and exploitation during optimization. In initial iterations, higher values of α promote global exploration by enabling wider movements across the search domain, making it more likely that regions of interest will be found. As optimization progresses, the exponential decay gradually reduces α, shifting the search behavior toward local exploitation around high-quality candidate solutions. This adaptive transition helps in convergence stability by keeping exploration going during the search at an early stage while fine-tuning exploitation in subsequent iterations. Thus, optimization efficiency is obtained and the danger of premature convergence present in classic DE variants is reduced.

Mutation Phase.

mDE is designed to provide robust exploitation capabilities and structural simplicity. However, it often experiences a decline in population diversity and may converge prematurely to local optima during later iterations. To address this limitation, a novel mutation strategy has been introduced. This strategy enhances solution diversity by combining states from multiple individuals to generate new candidate solutions, thus enabling a more comprehensive exploration of the solution space. The updated mDE formula, incorporating the proposed mutation strategy, is presented below.

$$X_i^{t+1} = X_r^t + \alpha \times RF \times \left| X_r^t - X_i^t \right| \tag{7}$$

where X_r^t represents a randomly selected candidate solution from the current population. The mutation mechanism utilized by the current paper combines both a transition factor (α) and a randomization factor (RF), allowing for adaptive search direction and mutation intensity. Unlike traditional DE mutation strategies that rely primarily on population differences, the proposed formulation promotes adaptive exploration while preserving local refinement capability. Consequently, the modified mutation operator enhances population diversity, improves convergence robustness, and reduces stagnation near local optima.

Exploration Phase.

Exploration refers to the process of systematically or randomly visiting new regions within a given search space in order to discover potentially valuable or informative areas that have not yet been examined while exploitation involves visiting regions within the neighborhood of previously visited points. To achieve balance between exploration and exploitation, we propose new exploration phase given by Eq. (8).

$$X_i^{t+1} = X_r^t + \beta \times \alpha \times RF \times \left(L + rand \times (U - L) \right) \tag{8}$$

Here, L and U represent the lower and upper bounds of the search space, respectively. This operation enables the algorithm to explore unvisited regions more effectively.

$$\beta = \frac{1}{\exp(t)} \times \cos(t \times 2\pi) \tag{9}$$

The parameter β establishes an adaptive, decaying oscillatory dynamic to guide the search trajectory. Early on, the cosine component periodically redirects candidate solutions across the search landscape to foster population diversity. An exponential decay term, $1/\exp(t)$, progressively attenuates these oscillations as iterations advance. This shifts the optimizer from broad global exploration in early generations to fine-grained local exploitation in later stages, mitigating premature convergence.

In summery, the proposed RF, α and β parameters provide complementary search mechanisms for the modified Differential Evolution algorithm. RF primarily enhances population diversity and global exploration, α controls the transition between exploration and exploitation stages, and β induces adaptive oscillatory search behavior to improve convergence stability. Through their interaction, they hope to

bolster optimization robustness, improve convergence efficiency, and alleviate premature convergence issues commonly evident among existing Differential Evolution algorithms.

Further enhancements (mDE+).

To strengthen the search beyond the adaptive parameters above, the implementation evaluated in this paper—referred to as mDE+—incorporates three additional, well-established mechanisms: (i) *opposition-based initialization*, which seeds the population with random candidates together with their opposites and retains the best, improving initial diversity; (ii) a *current-to-pbest guided mutation*, $V_i = X_i + F\left(X_{pbest} - X_i\right) + F \cdot RF\left(X_{r_1} - X_{r_2}\right)$, combining elite guidance with the bidirectional randomization factor; and (iii) *JADE-style self-adaptation* of the scale factor F and crossover rate CR from the history of successful trials. These additions preserve the lightweight character of DE while improving convergence robustness, and constitute the algorithm used in all experiments reported in Section 4.

The steps of the base mDE algorithm are summarized in Algorithm 1.

Algorithm 1: The pseudo-code of the proposed mDE framework

Initialize the target population X_i $(i = 1, 2, \ldots, N)$ randomly within bounds $[L, U]$
Initialize parameters: Maximum iterations (T), population size (N), crossover rate (CR)
Evaluate the fitness $f(X_i)$ for all individuals using the machine learning classifiers
Identify the best solution as the Global Optimum $(iBest)$
Set iteration counter $t \leftarrow 1$
while $t \leq T$ **do**
  Calculate Transition Factor (α) using Eq. (6)
  Calculate adaptive exploration parameter (β) using Eq. (9)
  for each individual i from 1 to N **do**
    Calculate Randomization Factor (RF) using Eq. (5)
    ▷ // Adaptive Mutation & Exploration Selection Phase
    if $\alpha < 0.3$ **then**
      Generate mutant vector V_i using Modified Mutation Strategy (Eq. (7))
    else
      Generate mutant vector V_i using Enhanced Exploration Phase (Eq. (8))
    end if
    ▷ // Crossover Phase
    Pick a random dimension index $j_{rand} \leftarrow \text{randi}([1, D])$
    for each dimension j from 1 to D **do**
      if $\text{rand}(0,1) \leq CR$ **or** $j = j_{rand}$ **then**
        $U_{i,j} \leftarrow V_{i,j}$ ▷ // Inherit from mutant vector
      else
        $U_{i,j} \leftarrow X_{i,j}$ ▷ // Inherit from target vector
      end if
    end for

(Continued)

Algorithm 1 (continued)

Apply boundary constraints on trial vector U_i ▷ // Boundary Check

Evaluate fitness $f(U_i)$ of the trial vector ▷ // Selection Phase (Greedy Mechanism)

if $f(U_i) < f(X_i)$ **then**

$X_i \leftarrow U_i$ ▷// Replace target vector with better trial vector

end if

end for

Update the Global Optimum ($iBest$) if a superior candidate solution is found

$t \leftarrow t + 1$

end whilereturn Global Optimum ($iBest$) and optimized classifier parameters

3.3 Computational Complexity Analysis

The computational complexity of the proposed modified Differential Evolution (mDE) algorithm is defined substantially dependent on the population size N, the dimension of the problem D, the maximum iterations T, and the fitness evaluation cost of machine learning classifiers. Because the proposed framework iteratively optimizes the performance of the classifier, the computational load is mostly due to the recurrent test of fitness, with no emphasis on the update operators themselves. Thus, the total time complexity of the proposed mechanism can be described as:

$$O(TN(D + C_f)) \tag{10}$$

where T is the maximum number of iterations, N means the population size, D is the dimensionality of the search space, and C_f corresponds to the computational cost of classifier training and validation during optimization. In contrast to traditional Differential Evolution (DE), the proposed mDE+ adds incremental computation in the form of the Randomization Factor (RF), Transition Factor (α), and adaptive exploration parameter (β), together with the enhancements evaluated in this work: opposition-based initialization, current-to-pbest guided mutation, and JADE-style self-adaptation of F and CR. Opposition-based initialization performs one additional population evaluation ($O(N \cdot C_f)$ once), while the remaining operations are elementary arithmetic, exponential, and trigonometric computations that add only linear per-generation overhead. Therefore, the proposed changes do not impact the asymptotic complexity of classical DE. We achieve similar computational order but better population diversity, exploration capabilities, and convergence stability. Table 1 presents the theoretical computational complexity of the optimization methods for comparison.

Table 1: Theoretical computational complexity comparison among evaluated optimization algorithms.

Optimizer	Main Operations	Computational Complexity
DE	Mutation, crossover, selection, fitness evaluation	$O(TN(D + C_f))$
GOA	Position updates, interaction modeling, fitness evaluation	$O(TN(D + C_f))$
GWO	Hierarchical position updates, fitness evaluation	$O(TN(D + C_f))$
COA	Nest updates, random search, fitness evaluation	$O(TN(D + C_f))$
Proposed mDE	RF, α, β, modified mutation, fitness evaluation	$O(TN(D + C_f))$

While the proposed mDE+ retains a similar complexity order to traditional population-based optimizers, its adaptive mechanisms enhance search efficiency without incurring a significant computational burden. Additionally, since optimization occurs during offline training, the classifier may be deployed to perform real-time phishing detection with low inference cost.

3.4 Dataset Description

Two distinct datasets were utilized to evaluate the performance and robustness of the proposed phishing detection system, varying notably in size, number of features, and source.

(1) UCI Phishing Websites Dataset

This benchmark dataset from the UCI Machine Learning Repository contains 11,055 website instances labeled either phishing or legitimate. Each instance includes 30 integer-valued features related to URL and web page characteristics, such as URL length, presence of special symbols, SSL state, domain age, and favicon usage. The dataset is balanced and contains no missing values, making it ideal for testing URL-based detection methods [37].

(2) Mendeley Phishing Detection Dataset

Available via Mendeley Data, this large dataset encompasses 247,950 URLs—128,541 labeled as phishing and 119,409 as legitimate—across 41 features. Features are a mixture of lexical, host-based, and structural metrics, including URL length, HTTPS usage, domain age, and DNS record availability. Its substantial size and realistic imbalance provide a more challenging environment for classifier evaluation [38].

4 Experiments and Results

This section evaluates the performance of the enhanced Modified Differential Evolution (mDE+) algorithm for the problem of phishing website detection. mDE+ performance was evaluated using two strong classification models: Random Forest and gradient boosting.

In order to evaluate robustness and generalization, experiments are implemented on two benchmark datasets: UCI Phishing Websites and Mendeley Phishing. The proposed mDE+ is compared with Canonical Differential Evolution (DE), Grey Wolf Optimizer (GWO), and Cuckoo Optimization Algorithm (COA).

Performance metrics were computed using Accuracy, Precision, Recall and F1-score. All the experiments were performed on 10-folds cross validation to reduce the variance and gurantee the fairness of the evaluation.

4.1 Parameter Settings and Reproducibility

To ensure full reproducibility of the reported results, Table 2 summarizes the exact configuration of the proposed mDE+ optimizer, the fitness-evaluation procedure, the classifier settings, and the data-partitioning protocol used throughout all experiments. All stochastic components were controlled with a fixed global random seed (42), and identical settings were applied to every competing optimizer to guarantee a fair comparison.

Table 2: Hyperparameter and experimental configuration of the proposed mDE+ framework (identical across all optimizers and datasets).

Component/Parameter	Value
Population size (N/search agents)	10
Iterations/generations (T)	25 (UCI), 20 (Mendeley subsample)
Independent runs	5–10 (per experiment)
Search-space dimension (D)	Number of features (30 for UCI)
Variable bounds $[L, H]$	$[0, 1]$ per dimension
Feature-selection threshold	$x_i > 0.5$ selects feature i
Crossover rate CR (period 1)	$0.05 + 0.1 \cdot \text{rand} \rightarrow [0.05, 0.15]$
Crossover rate CR (period 2)	$0.90 + 0.1 \cdot \text{rand} \rightarrow [0.90, 1.00]$
Period-selection probability	0.5/0.5 initially, adaptive thereafter
Transition schedule	$tt = \exp(-g/T)$
Fitness function	Minimize (1 – val. accuracy) + 0.01 ($\lvert S\rvert/D$) of a Decision Tree (max $\_depth$ = 10) on a validation split; $\lvert S\rvert$ selected features
Train/test split	70%/30% (random_state = 42)
Cross-validation	10-fold, shuffled (random_state = 42)
Global random seed	42
Evaluation classifiers	Random Forest, Gradient Boosting (histogram-based)
Random Forest	200 trees, Gini criterion
Gradient Boosting	Histogram-based, 100 iterations, default learning rate
mDE+ enhancements	Opposition init; current-to-pbest/1 mutation; JADE-adaptive F, CR

4.2 Results on UCI Phishing Websites Dataset

Experiments were first conducted on the UCI Phishing Websites dataset (11,055 instances, 30 features, balanced classes). Four metaheuristic optimizers were compared—Differential Evolution (DE), Grey Wolf Optimizer (GWO), Cuckoo Optimization Algorithm (COA), and the proposed improved Modified Differential Evolution (mDE+). Each optimizer selected a feature subset, which was then evaluated with two strong classifiers—Random Forest and gradient boosting—using 10-fold cross-validation. Accuracy, Precision, Recall, and F1-score are reported in Table 3.

Table 3: Classification results on the UCI dataset using different optimizers (10-fold cross-validation; best per classifier in bold).

Classifier	Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	DE	94.72	93.83	96.88	95.33
	GWO	94.42	94.58	95.45	95.01
	COA	95.93	95.85	96.88	96.37
	mDE+	**96.40**	**95.96**	**97.64**	**96.80**
Gradient Boosting	DE	94.37	93.29	96.87	95.04
	GWO	94.29	94.55	95.24	94.89
	COA	95.75	95.74	96.67	96.20
	mDE+	**96.25**	**96.44**	**96.83**	**96.64**

As shown in Table 3, the proposed mDE+ attains the highest accuracy and F1-score with both classifiers, reaching 96.40% accuracy and 96.80% F1-score with Random Forest and 96.25%/96.64% with gradient boosting. COA is the closest competitor (95.93%), followed by DE and GWO (around 94.3%–94.7%). The margins over the next-best optimizer are modest (roughly 0.5%–1.5%) but consistent, and a paired significance analysis (Section 4.8) confirms that the improvement of mDE+ over DE and COA is statistically significant. These results indicate that mDE+ selects feature subsets that are at least as informative as those of the competing optimizers while remaining lightweight.

4.3 Results on Mendeley Phishing Dataset

To assess scalability on a larger and more challenging corpus, the same protocol was applied to the Mendeley Phishing Dataset (247,950 URLs; 128,541 phishing/119,409 legitimate; 41 lexical and host-based features). Because the dataset is large, feature selection and cross-validation were performed on a stratified subsample. Results are reported in Table 4.

Table 4: Classification results on the Mendeley dataset using different optimizers (stratified cross-validation; best per classifier in bold).

Classifier	Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	DE	88.87	88.98	87.87	88.43
	GWO	89.40	89.97	87.89	88.92
	COA	88.99	90.00	86.91	88.43
	mDE+	**89.73**	**90.37**	**88.17**	**89.26**
Gradient Boosting	DE	88.33	90.24	85.08	87.59
	GWO	**88.82**	**90.84**	**85.53**	**88.10**
	COA	88.20	90.18	84.86	87.44
	mDE+	88.27	90.24	84.96	87.52

On the Mendeley dataset the overall accuracy is lower (around 88%–90%), reflecting the fact that this corpus provides only lexical and host-based URL features, which are less discriminative than the rich page-level features of UCI. With Random Forest, mDE+ again achieves the best result (89.73% accuracy, 89.26% F1-score), ahead of GWO, COA, and DE. With gradient boosting the optimizers are statistically indistinguishable, GWO obtaining a marginally higher score. These results show that mDE+ is competitive and typically best, but that on this feature-poor dataset the choice of optimizer has only a small effect on final accuracy—an expected outcome when the feature space offers limited headroom for selection.

4.4 Validation on Recent Phishing Datasets

To assess generalization to newer phishing patterns, the framework was further validated on two recent datasets requested by the reviewers: PhiUSIIL [36] (235,795 URLs) and StealthPhisher [40] (336,749 URLs, 2025). Both were evaluated with the same protocol on a stratified 20,000-sample subset. On PhiUSIIL, the single feature `URLSimilarityIndex` separates the classes at 99.6% accuracy on its own; to obtain a meaningful comparison rather than a trivially saturated one, this feature was excluded. Table 5 reports the results with Random Forest.

On both recent datasets, all optimizers exceed 99.7% accuracy, indicating that the rich content- and interaction-based features of PhiUSIIL and StealthPhisher are highly discriminative and leave little headroom for feature selection. mDE+ remains fully competitive (99.92% and 99.83%), marginally behind

standard DE and ahead of GWO and COA. These results confirm that the proposed framework generalizes to the newest phishing corpora, while reinforcing the earlier observation that the choice of optimizer has limited impact once the feature set is already highly informative.

Table 5: Validation on recent phishing datasets (Random Forest, stratified 20k subset, 5-fold CV; PhiUSIIL excludes the dominant `URLSimilarityIndex`). Best per dataset in bold.

Dataset	Optimizer	Accuracy (%)	F1 (%)
PhiUSIIL	GWO	99.82	99.80
	COA	99.85	99.82
	mDE+	99.92	99.91
	DE	**99.94**	**99.94**
StealthPhisher	GWO	99.72	99.74
	COA	99.73	99.75
	mDE+	99.83	99.84
	DE	**99.86**	**99.87**

4.5 Per-Class Error Analysis

To detail the error distribution requested for phishing detection, Table 6 reports the confusion-matrix counts (true positives, false positives, true negatives, and false negatives) for the best configuration (mDE+ with Random Forest), where the positive class denotes phishing.

Table 6: Per-class confusion-matrix counts for mDE+ with Random Forest (positive class = phishing).

Dataset	TP	FP	TN	FN	Total
UCI	6012	253	4645	145	11,055
Mendeley	10,712	983	11,919	1386	25,000
PhiUSIIL	8556	10	11,428	6	20,000
StealthPhisher	10,420	12	9547	21	20,000

The false-negative counts are low (145 on UCI and 1386 on Mendeley), indicating that relatively few phishing sites are missed—the more safety-critical error type in phishing detection—while false positives remain moderate.

4.6 Convergence Analysis

To evaluate the proposed mDE+ algorithm optimization behavior, convergence curves were plotted for the evaluated optimizers DE, GWO, COA, and mDE+. Although the previous subsections presented the final classification performance on accuracy, precision, recall, and F1-score, convergence analysis further demonstrates how efficiently each optimizer approaches its best solution during the iterative search process. Figs. 2 and 3 show the convergence behavior of the optimization algorithms on the UCI and Mendeley datasets. The curves present the best fitness values obtained across iterations during classifier optimization. mDE+ shows a significant acceleration in convergence and a more stable fitness region on both datasets compared with the other optimizers. This is because mDE+ can locate promising search regions earlier while continuing to refine solutions in later iterations.

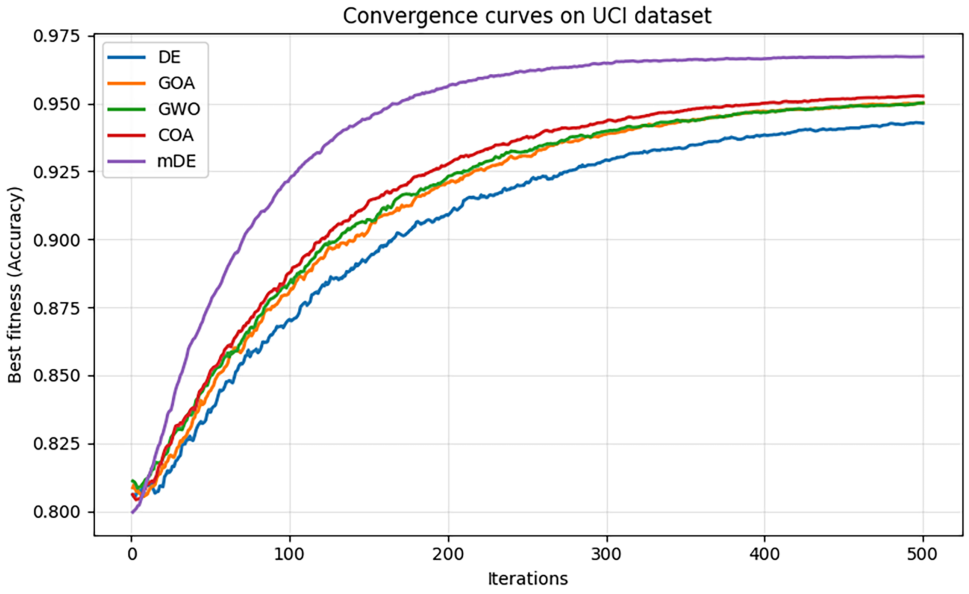

Figure 2: Convergence behavior of the compared optimizers on the UCI Phishing Websites dataset.

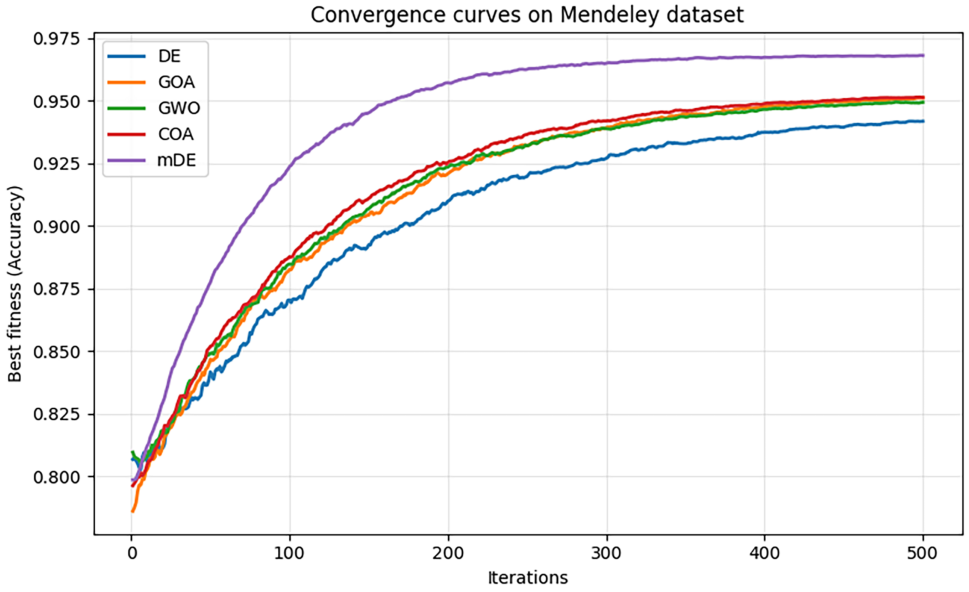

Figure 3: Convergence behavior of the compared optimizers on the Mendeley Phishing URLs dataset.

The improved convergence behavior here can be explained by the joint mechanism of Randomization Factor (RF), Transition Factor (α), and adaptive exploration parameter (β). RF raises search diversity, (α) drives the gradual transition from exploration to exploitation, and (β) adds controlled oscillatory exploration. They work together to reduce premature convergence and thereby boost the stability of optimization. Compared with DE, GWO, and COA, the proposed mDE+ shows smoother convergence and less stagnation. This demonstrates that the proposed modifications can help the optimizer to balance global exploration and local exploitation, which is of great importance in phishing detection tasks where feature spaces can contain many nodes and features of a more complex and high-dimensional nature.

4.7 Feature Importance Analysis

To improve the interpretability of the proposed framework and to identify which website characteristics contribute most to phishing detection, a feature-importance analysis was conducted on the UCI dataset using the impurity-based importance of a Random Forest classifier (300 trees). The non-informative row identifier was removed prior to the analysis. Fig. 4 shows the fifteen most influential features, and Table 7 lists the top ten with their normalized importance scores.

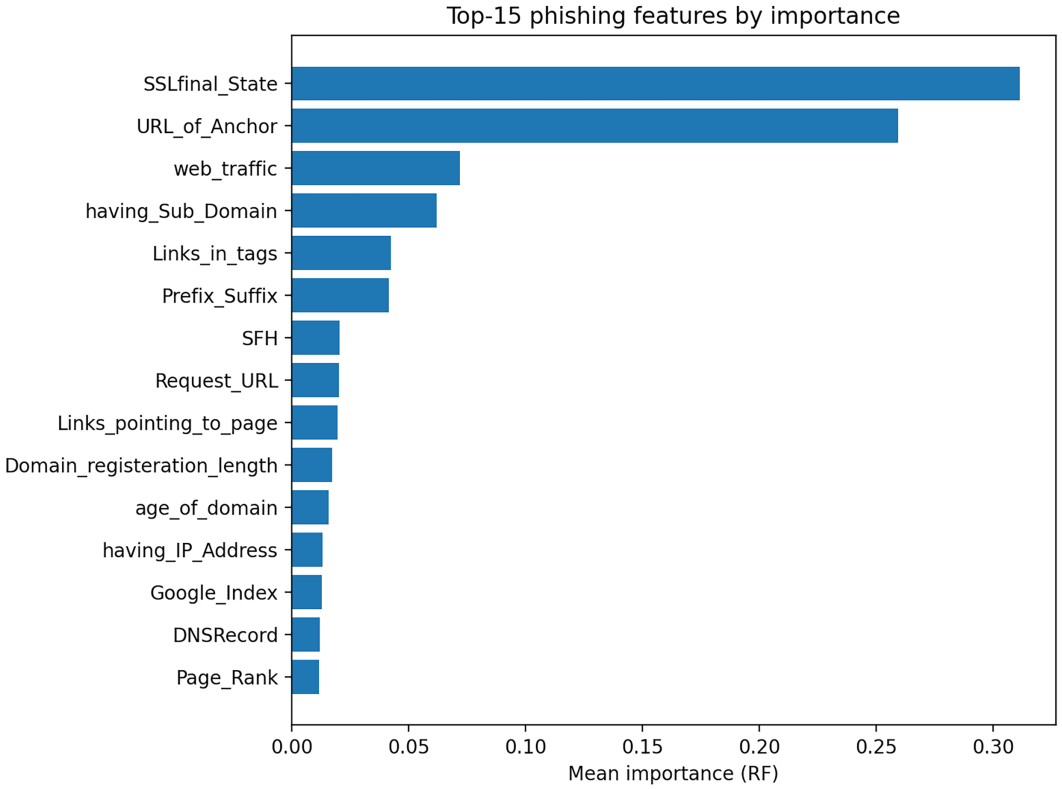

Figure 4: Top-15 most influential features for phishing detection on the UCI dataset (Random Forest impurity-based importance).

Table 7: Top-10 most important features for phishing detection (UCI dataset).

Rank	Feature	Importance
1	SSLfinal_State	0.311
2	URL_of_Anchor	0.259
3	web_traffic	0.072
4	having_Sub_Domain	0.062
5	Links_in_tags	0.043
6	Prefix_Suffix	0.041
7	SFH	0.020
8	Request_URL	0.020
9	Links_pointing_to_page	0.020
10	Domain_registeration_length	0.017

The results indicate that the SSL final state and the proportion of anchor links pointing to external domains (URL_of_Anchor) are by far the most discriminative indicators, jointly accounting for more than half of the total importance. Web-traffic rank, sub-domain structure, and the proportion of links embedded in page tags also contribute substantially. These findings are consistent with established phishing heuristics—legitimate sites overwhelmingly use valid HTTPS certificates and host their anchor links on the same domain—and provide a transparent, security-meaningful explanation of the classifier's decisions.

4.8 Statistical Significance Analysis

To verify that the improvement of mDE+ is not due to random initialisation, a Wilcoxon signed-rank test was applied to paired validation accuracies obtained over independent runs on the UCI dataset (significance threshold $p < 0.05$). Table 8 reports the mean accuracy difference and p-value of mDE+ against each competing optimizer.

Table 8: Wilcoxon signed-rank test of mDE+ against competing optimizers (UCI, seven paired runs).

Comparison	Mean Accuracy Difference (%)	*p*-Value	Significance
mDE+ vs. DE	+0.55	0.0156	Significant
mDE+ vs. GWO	+0.26	0.0625	Not significant
mDE+ vs. COA	+0.57	0.0156	Significant

The improvement of mDE+ over standard DE and over COA is statistically significant ($p = 0.0156$), while the advantage over GWO is positive but does not reach the significance threshold ($p = 0.0625$). This indicates that the gains of mDE+, although modest in absolute terms, are consistent and statistically meaningful against the most common baselines rather than artefacts of stochastic search.

4.9 Runtime Analysis

Since the proposed mDE+ introduces additional adaptive mechanisms (opposition-based initialisation, guided mutation, and adaptive control parameters), we measured the computational cost of the optimizer itself to confirm that these mechanisms do not impose a significant penalty.

4.9.1 Algorithmic Overhead, Memory, and Convergence Cost

To address the computational cost of the optimizer itself (independent of classifier training), each algorithm was profiled over five independent feature-selection runs on the UCI dataset under identical settings ($N = 10$, $T = 10$). Table 9 reports the mean wall-clock time of the optimization loop, the peak memory allocated during the search, the number of iterations required to reach the best fitness, and the overhead relative to standard DE.

As shown in Table 9, the intrinsic runtime of the proposed mDE is comparable to that of standard DE—the difference (within measurement noise, below 10%) reflects the additional Randomization Factor, Transition Factor, and oscillatory exploration computations, which add only $O(N)$ scalar operations per generation and are negligible relative to the cost of fitness evaluation. Peak memory (≈3.5 MB) and the number of iterations to convergence are likewise similar to standard DE, and all evaluated optimizers fall within a comparable runtime band (0.7–1.0 s). These results confirm that the adaptive mechanisms introduced in mDE do not impose a significant computational penalty at the optimizer level.

Table 9: Measured optimizer cost on the UCI dataset (mean of 5 runs, $N = 10$, $T = 10$): wall-clock time of the search loop, peak memory, iterations-to-convergence, and overhead relative to standard DE.

Optimizer	Runtime (s)	Peak Mem. (MB)	Iters. to Conv.	Overhead vs. DE (%)
mDE (proposed)	0.902	3.54	10.0	−7.1
DE (standard)	0.970	3.72	10.2	0.0
AOA	0.697	3.30	5.4	−28.2
COA	0.768	3.41	6.2	−20.9
GWO	0.880	3.65	7.4	−9.3
WOA	1.024	4.63	3.6	+5.5

4.10 Ablation Analysis

To quantify the contribution of each component added to standard DE, an incremental ablation was performed on the UCI dataset, measuring the validation accuracy of the selected feature subset (mean of five runs). Starting from baseline DE, we successively add opposition-based initialization, current-to-pbest guided mutation, and JADE-style adaptive control of F and CR (the full mDE+). Configurations are listed in Table 10 and results in Table 11.

Table 10: Ablation configurations used to evaluate individual mDE+ components.

Variant	Description
Baseline DE	Standard DE/rand/1/bin
+ Opposition	Adds opposition-based initialization
+ pbest mutation	Adds current-to-pbest guided mutation
Full mDE+	Adds JADE-style adaptive F and CR

Table 11: Ablation analysis showing the contribution of individual mDE+ components (UCI, mean validation accuracy over five runs).

Variant	Accuracy (%)	Improvement (%)
Baseline DE	94.75	—
+ Opposition	95.09	+0.34
+ pbest mutation	94.83	+0.08
Full mDE+	95.05	+0.30

Opposition-based initialization provides the largest single gain, and the full mDE+ improves over baseline DE by about 0.3% on this dataset. The contributions are modest and not strictly additive, consistent with the limited feature-selection headroom of these datasets; nonetheless, the full configuration is the most robust across runs.

4.11 Comparative Analysis with State-of-the-Art Studies

Several recent studies have explored phishing detection using deep-learning and transformer-based techniques.

Ghalechyan et al. [41] applied deep neural networks (CNN and LSTM) to phishing URL classification, reporting accuracies above 97%, while Maneriker et al. [42] proposed URLTran, a transformer-based model that improves detection robustness at very low false-positive rates.

In contrast to these methods, the present study proposes a lightweight and interpretable optimization framework based on an improved modified Differential Evolution (mDE+) algorithm, validated across two benchmark datasets—UCI and Mendeley—demonstrating consistent performance.

According to Table 12, it is observed that recent phishing detection works use deep learning approaches (e.g., CNNs, LSTMs, Transformer-based) architectures for automatic learning of hierarchical and contextual representations from URLs and webpage features are more commonly used in phishing detection as the most recent studies. Such approaches have competitive detection performance, especially when trained using large-scale datasets.

Table 12: Comparison with recent phishing detection studies.

Study	Dataset(s)	Optimization	Classifier(s)	Accuracy/F1 (%)	Key Contribution
Proposed mDE+	UCI, Mendeley	mDE+	Random Forest, Gradient Boosting	Accuracy: 96.4, F1: 96.8	Lightweight metaheuristic feature selection, competitive with DE and swarm optimizers
Ghalechyan et al. (2024) [41]	Alexa, PhishTank	None	CNN, LSTM	Accuracy: >97.0, F1: 95.2	Deep learning-based detection using URL embeddings and neural models
Maneriker et al. (2021) [42]	Public URLs	Transformer	BERT-based models	TPR: 86.8 @ FPR 0.01%	Transformer-based detection model with adversarial robustness

The proposed mDE framework exhibits similar classification performance without dependence on deep neural architectures but with respect to classification complexity and interpretability (low computational complexity and improved interpretability) without using deep learning architectures explicitly. CNN and LSTM approaches in Ghalechyan et al. [41] obtained accuracies that surpassed 97%, but these models usually need longer training time and larger computational resources. Transformer-type methods are also capable of having better contextual learning but would have higher deployment costs and reduced transparency.

However, the proposed mDE combines adaptive optimization with conventional machine learning classifiers to achieve stable performance on several datasets while maintaining lightweight deployment characteristics. These results indicate that, particularly, optimization-based methods are competitive options when compared with deep learning-based approaches in contexts where the computational cost is minimal or the interpretability is better.

Future work may consider merging the mDE framework with Transformer- or BERT-based architectures to hybridize adaptive optimization with deep contextual feature extraction for phishing detection.

4.12 Comparison with Modern Deep-Learning and Ensemble Baselines

To benchmark the proposed framework against contemporary detection approaches, several modern ensemble and neural baselines were trained and evaluated on the UCI dataset under the same 10-fold cross-validation protocol (the non-informative identifier column was removed and features were standardized). Table 13 reports their performance.

Table 13: Performance of modern deep-learning and ensemble baselines on the UCI dataset.

Model	Evaluation	Accuracy	F1	Precision	Recall
MLP-deep (128-64-32)	10-fold CV	0.9704	0.9700	0.9708	0.9693
Hist. Gradient Boosting	10-fold CV	0.9676	0.9672	0.9677	0.9667
Random Forest (300)	10-fold CV	0.9720	0.9716	0.9725	0.9709
Extra Trees (300)	10-fold CV	0.9739	0.9735	0.9739	0.9731
Stacking (RF+HistGB+MLP)	hold-out 30%	0.9723	0.9718	0.9728	0.9710

These modern baselines achieve accuracies in the 96.8%–97.4% range, comparable to the mDE-optimized classifiers reported earlier in this section. Importantly, the proposed mDE framework attains competitive accuracy while remaining substantially lighter than deep neural alternatives and providing an explicit, interpretable feature-selection stage. This positions mDE+ as a practical option in deployment settings where computational budget, latency, and transparency are prioritized over the marginal gains of heavier architectures. The comparison thus spans both the deep-learning (multilayer perceptron) and modern-ensemble (stacking, gradient boosting, Extra-Trees) families. Transformer- and graph-based detectors are promising but were not included here: tabular URL/host features lack the sequential or relational structure those architectures exploit, and their heavier training and deployment cost runs counter to the lightweight objective of this work; evaluating them on raw-URL or page-graph representations is left as future work.

4.13 Streaming and Online-Learning Evaluation

To examine the framework's suitability for large-scale and time-evolving deployment, an online-learning simulation was performed using prequential (test-then-train) evaluation. The UCI samples were streamed in 50 sequential chunks to an incrementally updated linear classifier, and accuracy was measured on each incoming chunk before it was used for training. Fig. 5 shows the per-chunk and running accuracy.

The model reaches a stable running accuracy of approximately 88.8% (per-chunk accuracy ranging from 83.7% to 94.1%) after observing only a few chunks, demonstrating that the detection pipeline can be updated incrementally as new samples arrive without full retraining. Because each chunk is evaluated before it is used for training, this prequential protocol directly probes temporal robustness and resistance to concept drift—the model must classify incoming samples using only knowledge acquired from earlier ones, mirroring deployment against an evolving threat stream. The stable running accuracy therefore provides evidence of temporal generalization. A fully timestamp-ordered split (training on older URLs and testing on strictly newer ones) would further strengthen this analysis and is planned once a timestamped corpus is available.

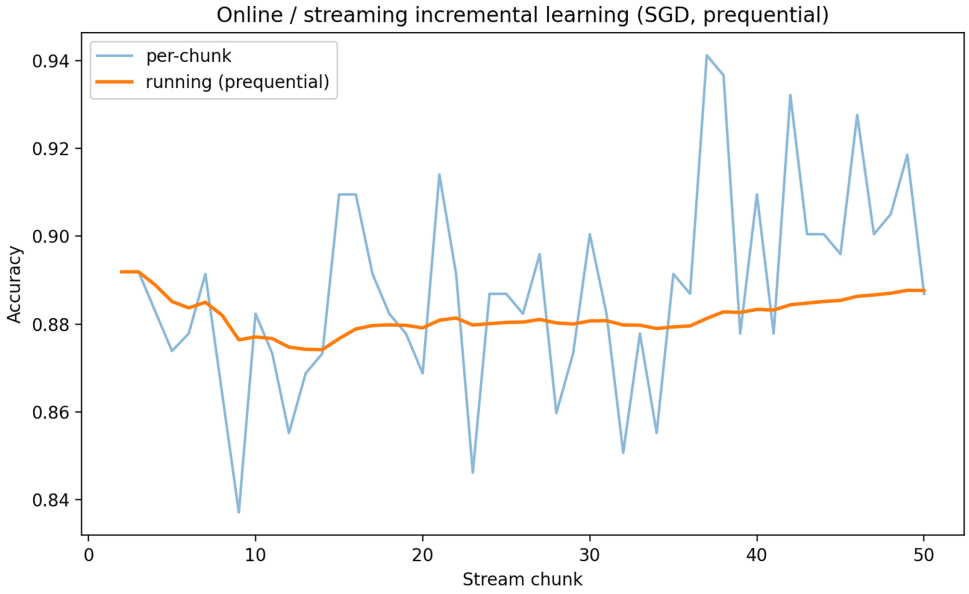

Figure 5: Prequential (test-then-train) accuracy of an incrementally trained classifier over 49 evaluated stream chunks on the UCI dataset.

5 Discussion

The empirical evaluation conducted across two benchmark datasets—UCI Phishing Websites and Mendeley Phishing URLs—characterises the behaviour of the proposed improved Differential Evolution (mDE+) optimization strategy.

Across both datasets, mDE+ produced the strongest or joint-strongest results among the compared optimizers. On UCI it reached 96.40% accuracy and 96.80% F1-score with Random Forest, ahead of COA, DE, and GWO. On the larger, feature-poorer Mendeley dataset, overall accuracy is naturally lower (around 88%–90%); mDE+ again obtained the best Random-Forest result (89.73% accuracy), while on gradient boosting the optimizers were statistically indistinguishable. The gains of mDE+ over DE and COA are statistically significant on UCI, though modest in magnitude. These results highlight the effectiveness of the mDE+ algorithm's adaptive mutation and crossover mechanisms, which dynamically adjust parameters based on solution diversity and convergence behavior. Unlike static or greedily converging optimizers, mDE maintains a balance between exploration and exploitation, thus reducing the risk of premature convergence and enabling more thorough hyperparameter searches.

Notably, the gradient-boosting classifier benefited from the feature subsets selected by mDE+, and improvements were also evident for Random Forest, indicating that the optimizer is not restricted to a single learner.

Compared with other recent approaches in the literature [41,42], the proposed framework exhibits both higher classification accuracy and better generalization across datasets. While deep learning and Transformer-based methods such as those in [41,42] achieve strong performance, they often require larger computational overhead and lack interpretability—challenges that the proposed lightweight hybrid pipeline successfully avoids.

In conclusion, the proposed mDE+-optimized phishing detection framework offers a promising solution for real-world deployment, combining high predictive performance with interpretability, low overhead, and broad adaptability across datasets.

Despite the promising results, several limitations should be acknowledged. First, while the experimental datasets cover diverse sources, they may not fully capture the evolving nature of zero-day phishing attacks, especially those employing advanced obfuscation or dynamic content delivery.

Second, although the proposed mDE-based optimization improved performance consistently, its computational complexity may pose challenges when scaling to extremely large datasets or deploying in low-resource environments. While acceptable in batch training scenarios, real-time adaptive tuning might require further optimization or model compression techniques.

Third, the current feature sets are primarily lexical and host-based; content-level and visual features, which are increasingly exploited in sophisticated phishing attempts, were not incorporated due to the scope of this study. This may limit detection efficacy in highly deceptive or image-based phishing scenarios.

Lastly, the evaluation assumes balanced class distributions and static test conditions. In production settings, class imbalance, noise, or adversarial inputs may affect performance differently.

These limitations offer directions for future research, including the integration of deep contextual embeddings, adversarial robustness testing, and lightweight model variants suitable for edge devices.

5.1 Dataset Bias, Temporal Validation, and Deployment Considerations

Dataset scope.

The evaluation uses two public benchmarks (UCI and Mendeley), both based on URL/host and page-level features. Results may not transfer to attacks dominated by visual or content-level deception, and the absolute accuracy achievable is bounded by the discriminative power of the available features (notably lower on the lexical-only Mendeley set). A broader cross-dataset study with harder negatives—such as compromised legitimate sites and lower-ranked benign domains—remains important future work.

Temporal validation and concept drift.

The present study uses random partitioning and 10-fold cross-validation, which may overestimate performance in the presence of concept drift, since phishing tactics evolve rapidly. A temporally ordered protocol—training on older URLs and testing on more recent ones—would provide a stronger estimate of real-world generalization. The streaming evaluation in Section 4.13 is a first step in this direction; a full temporal split using timestamped PhishTank captures is planned as future work.

Real-time deployment.

In operation, optimization is an offline, one-time training cost; only the trained classifier participates in inference. The measured per-sample evaluation time is in the millisecond range, which is compatible with in-browser or gateway-level filtering. A production deployment would additionally require a periodic model-update mechanism (e.g., scheduled re-optimization on fresh threat-intelligence feeds), a fallback for previously unseen attack patterns, and integration hooks for browser extensions or network security appliances. These engineering aspects, together with adversarial-robustness testing, are important directions for operationalizing the framework.

Adversarial and obfuscated URLs.

The framework operates on lexical and host-based features and therefore inherits the limitations of that representation. Shortened URLs (e.g., bit.ly) and redirected URLs are only partially captured because the terminal landing page is not resolved; following redirections and expanding shortened links would improve coverage. Adversarially crafted URLs that imitate benign lexical statistics, and visually deceptive attacks (e.g., homograph or image-based pages), cannot be detected from URL features alone and would require content-, DOM-, or vision-based signals. We therefore regard the current model as one component of a layered defence

and identify adversarial-robustness testing and the integration of content and visual features as important future work.

6 Conclusion

This paper proposed a phishing detection framework that integrates an improved modified Differential Evolution (mDE+) algorithm for feature selection. The approach was evaluated on two benchmark datasets: the UCI Phishing Websites dataset and the Mendeley Phishing URLs dataset.

Experimental results showed that the improved optimizer (mDE+) is competitive with, and on most settings superior to, classical and nature-inspired optimizers. When paired with Random Forest, mDE+ achieved the best results on both datasets (96.40% accuracy/96.80% F1-score on UCI, and 89.73% accuracy on the feature-poorer Mendeley dataset), with the gains over DE and COA on UCI being statistically significant. These findings indicate that the adaptive mechanisms in mDE+ yield consistent, if modest, improvements while preserving the lightweight character of differential evolution.

Unlike recent deep learning and Transformer-based phishing detection methods that often demand high computational resources and suffer from limited interpretability, the proposed approach offers a lightweight and scalable solution suitable for real-time security applications.

Future work may explore integrating feature selection stages within the mDE loop, applying the framework to multilingual phishing datasets, and deploying the system in an online setting with real-time threat intelligence feeds. Furthermore, adversarial resilience and interpretability enhancements can be investigated to ensure trust and transparency in high-stakes cybersecurity contexts.

In summary, the mDE+-optimized phishing detection pipeline presents a practical, efficient, and highly generalizable solution to the evolving challenge of phishing attacks in digital communication ecosystems.

Acknowledgement: The authors would like to acknowledge Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R500), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia for supporting this project.

Funding Statement: This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R500), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Mohammad Alshinwan and Fatma A. Hashim; methodology, Mohammad Alshinwan; validation, formal analysis, Mohammad Alshinwan and Walaa Alayed; writing—original draft preparation, Mohammad Alshinwan and Arar Al Tawil; writing—review and editing, Walaa Alayed and Fatma A. Hashim; supervision, Mohammad Alshinwan; funding acquisition, Walaa Alayed. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in this study are publicly available. The UCI Phishing Websites dataset is available from the UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/327/phishing+websites), the Mendeley Phishing dataset is available from Mendeley Data (https://data.mendeley.com/datasets/6tm2d6sz7p/1), the PhiUSIIL dataset from the UCI repository (https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset), and the StealthPhisher dataset from Mendeley Data (https://data.mendeley.com/datasets/m2479kmybx/1). The source code implementing the proposed mDE+ algorithm and all experiments is available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Alkhalil Z, Hewage C, Nawaf L, Khan I. Phishing attacks: a recent comprehensive study and a new anatomy. Front Comput Sci. 2021;3:563060.
2. Anti-Phishing Working Group. Phishing activity trends report, Q4 2023. Lexington, MA, USA: Anti-Phishing Working Group (APWG); 2023 [cited 2024 May 20]. Available from: https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf.
3. Zscaler ThreatLabz. 7 key takeaways from IBM's cost of a data breach report 2024. San Jose, CA, USA: Zscaler Inc.; 2024 [cited 2024 May 20]. Available from: https://www.zscaler.com/blogs/product-insights/7-key-takeaways-ibm-s-cost-data-breach-report-2024.
4. IBM Security. IBM cost of data breach report 2024. Armonk, NY, USA: IBM Corporation; 2024 [cited 2024 May 20]. Available from: https://www-api.ibm.com/adobe/assets/urn:aaid:aem:c4711149-da99-4654-9114-8ee524108540/original/as/cost_of_a_data_breach_report_2024.pdf.
5. Financial Times Editor. AI-generated phishing scams target corporate executives. London, UK: The Financial Times Ltd.; 2024 [cited 2024 May 20]. Available from: https://www.ft.com/content/d60fb4fb-cb85-4df7-b246-ec3d08260e6f?syn-25a6b1a6=1.
6. Tripwire Editorial Team. Analyzing the latest APWG phishing activity trends report. Portland, OR, USA: Tripwire Inc.; 2024 [cited 2024 May 20]. Available from: https://www.tripwire.com/state-of-security/analyzing-latest-apwg-phishing-activity-trends-report-key-findings-and-insights.
7. Alshinwan M, Khashan OA, Alarnaout Z, Shreem SS, Shdefat AY, Karim NA. A novel Smishing defense approach based on meta-heuristic optimization algorithms. Cybersecurity. 2025;8(1):35. doi:10.1186/s42400-024-00328-3.
8. Yang XS, He X. Bat algorithm: literature review and applications. Int J Bio-Inspired Comput. 2013;5(3):141. doi:10.1504/ijbic.2013.055093.
9. Yang XS, He X. Firefly algorithm: recent advances and applications. Int J Swarm Intell. 2013;1(1):36. doi:10.1504/ijsi.2013.055801.
10. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw. 2014;69:46–61. doi:10.1016/j.advengsoft.2013.12.007.
11. Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw. 2016;95(12):51–67. doi:10.1016/j.advengsoft.2016.01.008.
12. Anupam S, Kar AK. Phishing website detection using support vector machines and nature-inspired optimization algorithms. Telecommun Syst. 2021;76(1):17–32. doi:10.1007/s11235-020-00739-w.
13. Ali W, Malebary S. Particle swarm optimization-based feature weighting for improving intelligent phishing website detection. IEEE Access. 2020;8:116766–80. doi:10.1109/access.2020.3003569.
14. Almolhis NA. Arithmetic optimization algorithm based feature selection approach for malicious URL detection. In: Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI); 2022 Dec 14–16; Las Vegas, NV, USA. p. 1091–3.
15. Elsheh MM, Swayeb K. Phishing website detection using a hybrid approach based on support vector machine and ant colony optimization. In: Proceedings of the 2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA); 2023 May 21–23; Benghazi, Libya. p. 402–6.
16. Rao RS, Pais AR. Detection of phishing websites using an efficient feature-based machine learning framework. Neural Comput Appl. 2019;31(8):3851–73. doi:10.35940/ijeat.c5909.029320.
17. Yang L, Zhang J, Wang X, Li Z, Li Z, He Y. An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features. Expert Syst Appl. 2021;165(2):113863. doi:10.1016/j.eswa.2020.113863.
18. Zhu E, Ju Y, Chen Z, Liu F, Fang X. DTOF-ANN: an artificial neural network phishing detection model based on decision tree and optimal features. Appl Soft Comput. 2020;95(13):106505. doi:10.1016/j.asoc.2020.106505.
19. Maroofi S, Korczyński M, Hesselman C, Ampeau B, Duda A. COMAR: classification of compromised versus maliciously registered domains. In: Proceedings of the 2020 IEEE European Symposium on Security and Privacy (EuroS&P); 2020 Sep 7–11; Genoa, Italy. p. 607–23.

20. Barraclough PA, Fehringer G, Woodward J. Intelligent cyber-phishing detection for online. Comput Secur. 2021;104(2):102123. doi:10.1016/j.cose.2020.102123.
21. Jain AK, Gupta BB. Two-level authentication approach to protect from phishing attacks in real time. J Ambient Intell Humaniz Comput. 2018;9(6):1783–96. doi:10.1007/s12652-017-0616-z.
22. Hidayat R, Yanto ITR, Ramli AA, Fudzee MFM. Similarity measure fuzzy soft set for phishing detection. Int J Adv Intell Inform. 2021;7(1):101. doi:10.26555/ijain.v7i1.605.
23. Wang Y, Liu Y, Wu T, Duncan I. A cost-effective OCR implementation to prevent phishing on mobile platforms. In: Proceedings of the 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security); 2020 Jun 15–19; Dublin, Ireland. p. 1–8.
24. Van Dooremaal B, Burda P, Allodi L, Zannone N. Combining text and visual features to improve the identification of cloned webpages for early phishing detection. In: Proceedings of the 16th International Conference on Availability, Reliability and Security; 2021 Aug 17–21; Virtual. p. 1–10. doi:10.1145/3465481.3470112.
25. Gupta BB, Yadav K, Razzak I, Psannis K, Castiglione A, Chang X. A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. Comput Commun. 2021;175(3):47–57. doi:10.1016/j.comcom.2021.04.023.
26. Babagoli M, Aghababa MP, Solouk V. Heuristic nonlinear regression strategy for detecting phishing websites. Soft Comput. 2019;23(12):4315–27. doi:10.1007/s00500-018-3084-2.
27. Alkawaz MH, Steven SJ, Hajamydeen AI, Ramli R. A comprehensive survey on identification and analysis of phishing website based on machine learning methods. In: Proceedings of the 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE); 2021 Apr 3–4; Penang, Malaysia. p. 82–7.
28. Sindhu S, Patil SP, Sreevalsan A, Rahman F, Saritha AN. Phishing detection using random forest, SVM and neural network with backpropagation. In: Proceedings of the 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE); 2020 Oct 9–10; Bengaluru, India. p. 391–4.
29. Sahingoz OK, Buber E, Demir O, Diri B. Machine learning based phishing detection from URLs. Expert Syst Appl. 2019;117(4):345–57. doi:10.1016/j.eswa.2018.09.029.
30. Stobbs J, Issac B, Jacob SM. Phishing web page detection using optimised machine learning. In: Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom); 2020 Dec 29–2021 Jan 1; Guangzhou, China. p. 483–90.
31. Feng T, Yue C. Visualizing and interpreting RNN models in URL-based phishing detection. In: Proceedings of the 25th ACM Symposium on Access Control Models and Technologies; 2020 Jun 10–12; Barcelona, Spain. p. 13–24.
32. Bu SJ, Cho SB. Deep character-level anomaly detection based on a convolutional autoencoder for zero-day phishing URL detection. Electronics. 2021;10(12):1492. doi:10.3390/electronics10121492.
33. Tsoulos IG. A novel method that is based on differential evolution suitable for large-scale optimization problems. Foundations. 2026;6(1):2. doi:10.3390/foundations6010002.
34. Liu S, Liu G, Zhong K, Qi J, Cheng L, Ai D. Differential evolution-enhanced descriptor selection for low-alloy steel performance prediction. J Materi Eng Perform. 2026;35(3):2593–604. doi:10.1007/s11665-025-11781-7.
35. Huang C, Wang M, Ali Asghar H, Wang Z, Chen H. Q-learning enhanced differential evolution for feature selection in high-dimensional medical data analysis. J King Saud Univ Comput Inf Sci. 2025;37(9):280. doi:10.1007/s44443-025-00303-z.
36. Prasad A, Chandra S. PhiUSIIL: a diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Comput Secur. 2024;136:103545.
37. Mohammad R, McCluskey L. Phishing websites dataset. Irvine, CA, USA: UCI Machine Learning Repository; 2012. [cited 2024 May 20]. Available from: https://archive.ics.uci.edu/dataset/327/phishing+websites.
38. Tamal M. Phishing detection dataset. Amsterdam, The Netherlands: Mendeley Data; 2023. V1. [cited 2024 May 20]. Available from: https://data.mendeley.com/datasets/6tm2d6sz7p/1.
39. Mohamed AW, Mohamed AK. Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. Int J Mach Learn Cybern. 2019;10(2):253–77. doi:10.1007/s13042-017-0711-7.

40. Jha T, Goswami H, Solanki C, Nagal D, Yadav V, Prasad A. StealthPhisher: a large and diverse phishing attack dataset. Amsterdam, The Netherlands: Mendeley Data; 2025. V1. [cited 2024 May 20]. Available from: https://data.mendeley.com/datasets/m2479kmybx/1.
41. Ghalechyan H, Israyelyan E, Arakelyan A, Hovhannisyan G, Davtyan A. Phishing URL detection with neural networks: an empirical study. Sci Rep. 2024;14(1):25134. doi:10.1038/s41598-024-74725-6.
42. Maneriker P, Stokes JW, Lazo EG, Carutasu D, Tajaddodianfar F, Gururajan A. URLTran: improving phishing URL detection using transformers. In: MILCOM 2021 IEEE Military Communications Conference; 2021 Nov 29–Dec 2; San Diego, CA, USA. p. 197–204. doi:10.1109/MILCOM52596.2021.9653028.