Phishing Detection Method Using a Hybrid Method of Genetic Algorithm and particle Swarm Optimization

hamid muhamad eubayd
General Directorate of Education of Al-Rusafa II

1. Introduction

Phishing denotes a series of cyberattacks in which an attacker utilizes deception and disguise to obtain the victim's sensitive information online. Phishing has a historical precedent, with its inaugural documented instance being in 1995, when hackers impersonated American Online (AOL) personnel to deceive AOL customers into disclosing their usernames and passwords [AOHell] (Alabdan, 2020). During that period, hackers exploited instant messaging networks to deceive AOL users into divulging their usernames and passwords through direct messages. A distinguishing feature of Phishing compared to other cyberattacks is the incorporation of deceptive tactics in its strategy, implementation, and overall effectiveness. Phishing comprises a four-step procedure that entails the following [1]. Establishing trust which is the perpetrator cultivates the victim's Trust to elicit their requisite acts. Actions may include clicking webpages, responding to emails, etc. Trust is acquired through spoofed websites, email addresses, applications, and similar methods [2]. The second is the redirection which is an intermediary phase may be present or absent in various phishing assaults. A fake email may route the user to a phishing site using a link. Typically, activities following the establishment of trust lead to redirection. Users subsequently submit their credential information on the redirected sites or channels [3]. Acquiring data is later noticed that this phase marks the commencement of the attack. The assailant obtains the necessary information through misdirected forms or websites or as responses to spoofed emails from the victim. In the execution the perpetrator implements the necessary identity or financial fraud utilizing the identification or credential information acquired in Step 3 [4].

In the past two decades, phishing assaults have profoundly impacted enterprises worldwide [5]. As countermeasures against phishing attacks have evolved, so have the attackers, who have developed sophisticated methods for executing novel phishing assaults. Despite extensive studies conducted by industry and academia over several years, the threat of phishing assaults remains prevalent today [6].

In recent years, phishers have predominantly targeted SaaS (Software as a Service) and webmail, comprising 33% of attacks across all industry sectors [7]. IBM determined that 27% of phishing attacks in 2018 targeted webmail services. Furthermore, it was observed that 29 percent of the corporate attacks examined by X-Force attributed the breach to a phishing email [8].

Symantec discovered that in the underground economy, "custom phishing page services" are priced between USD 3 and 12, suggesting that the costs associated with initiating a custom phishing attack are negligible. Research indicates that gift cards have become one of the predominant methods for scammers to liquidate their profits [7]. The FBI calculated that the victim loss from phishing in 2018 amounted to USD 48,241,748, impacting 26,379 individuals [9].

In 2018, the FBI documented around 100 complaints, predominantly affecting the healthcare, education, and air travel sectors, culminating in a total net loss of almost USD 100 million. This scheme utilized phishing emails to target employees and obtain their login credentials. Subsequently, these were utilized to infiltrate the payroll system, following which the phishers instituted regulations preventing employees from receiving notices on modifications to their accounts. The phisher subsequently altered account holders' direct debit details to redirect funds into their own account, which in this case involved a prepaid card [10].

The repercussions of phishing assaults are extensively experienced across various sectors, including healthcare and education, as well as among persons engaged in online gaming. An illustration is a phishing fraud designed to get user login credentials for Steam, a PC gaming platform, by presenting a "free skin giveaway" (Figure 3). The fraud commenced with a comment posted on a user's profile, which, when clicked, redirected the victim to the A phishing website with details about the giveaway and a counterfeit scrolling chat bar to create an illusion of authenticity. The victim was directed to "login via Steam," leading them to a counterfeit login interface that grabbed their information. The assault involved the creation of a valid Steam Guard code (i.e., two-factor authentication), which provided the phisher with access to the victim's account to sell products and further advance the fraud (see Figure 3). [11]. Massively multiplayer online games (MMOs) are frequently targeted by phishers due to the potential for "loot box" items to be sold on the online black market.

A recent instance of this phishing scheme specifically targeted the MMO Elder Scrolls Online [12].

Future Internet 2020, 12, x FOR PEER REVIEW Three out of thirty-seven individuals was directed to a counterfeit login interface where their credentials were recorded. The assault involved the creation of a valid Steam guard code (i.e., two-factor authentication), which provided the phisher with access to the victim's account for the purpose of selling things and further advancing the fraud (see Figure 3) [11]. Massively multiplayer online games (MMOs) are frequently targeted by phishers due to the potential for "loot box" items to be sold on the online black market. A recent instance of this phishing scheme specifically targeted the Phishing contributed to the inaugural successful cyber attack on a power grid, occurring in Ukraine in December 2015. IT personnel and network administrators from multiple organizations involved in electricity distribution for Ukraine were subjected to spear phishing attempts. The assault entailed a malevolent Microsoft Word document that prompted the activation of macros. Upon activation, the macro deployed the BlackEnergy3 malware onto the system, thereby establishing a backdoor for the attackers. This ultimately led to the effective deactivation of 30 substations, leaving 230,000 individuals without electricity for up to six hours. This case illustrates the potency and destructiveness of a meticulously orchestrated and effectively implemented phishing assault. Even trained IT workers cannot consistently spot these threats [13].

The preceding discussion demonstrates that phishing constitutes a significant issue that requires thorough comprehension for effective mitigation. This article examines various attributes from traditional, contemporary, and innovative phishing strategies, highlighting deficiencies in existing anti-phishing measures. This study aims to present a method that is used for the phishing attacks classification using the optimization methods. In specific the used method is a hybrid Genetic Algorithm and the Particle Swarm Optimization. This combination helps to reach to the best classification accuracy faster than using other methods.

2. *Optimization Methods*

In this section we explain the two algorithm used in this work which are the most famous optimization algorithms.

2.1 Genetic Algorithms

The genetic algorithm (GA) is an optimization and search methodology grounded in genetics and natural selection. A genetic algorithm enables

several people to evolve according to defined selection criteria. A condition that optimizes "fitness" (i.e., reduces the cost function). The method was conceived by John Holland in 1975 during the 1960s and 1970s and subsequently popularized by his student, David Goldberg, who successfully addressed a complex issue related to gas-pipeline transmission management for his dissertation in 1989. Holland's seminal work was encapsulated in his publication. He was the inaugural individual to attempt to establish a theoretical foundation for genetic algorithms through his schema theorem. De Jong's (1975) research demonstrated the efficacy of genetic algorithms for function optimization and was the initial systematic attempt to identify optimum parameters for genetic algorithms. Goldberg has likely provided the most impetus to the GA movement through his successful applications and exemplary book (1989). Since then, numerous iterations of evolutionary programming have been attempted with differing levels of success. Advantages of a Genetic Algorithm (GA) include its ability to [14]:

- Optimize both continuous and discrete variables,
- Operate without requiring derivative information,
- Conduct simultaneous searches across a broad sampling of the cost surface,
- Manage a substantial number of variables,
- Function effectively on parallel computing systems,
- Optimize variables characterized by highly complex cost surfaces (capable of escaping local minima).
- Generate a list of optimal variables rather than a singular solution.
- Potentially encode variables for optimization using these encoded representations, and
- Utilize numerically generated data, experimental data, or analytical functions.

These advantages are compelling and yield remarkable outcomes when conventional optimization methods falter significantly.

There are better solutions than the GA for some issues. For example, conventional approaches have been optimized to rapidly identify the solution of a well-behaved convex analytical function with a limited number of variables. In such instances, calculus-based solutions surpass the genetic algorithm by quickly identifying the minimum, while the genetic algorithm still evaluates the initial population's costs. The optimizer should leverage historical experience and utilize these expedited approaches for these issues.

Nonetheless, numerous practical issues need to conform to this classification. Furthermore, alternative methods may yield solutions more rapidly than the Genetic Algorithm for relatively easy situations. The extensive population of solutions that endows the genetic algorithm with its efficacy also hinders its speed on a serial computer, as the cost function for each solution must be assessed. Nevertheless, if a parallel computer is accessible, each processor can evaluate a distinct function concurrently. Consequently, the GA is ideally configured for parallel computations [15].

2.2 Genetic Algorithm Implementation

The genetic algorithm is implemented by first initializing the candidate solutions and then applying the meta heuristic steps until a specified number of iterations. The steps are represented in figure 2 [16].

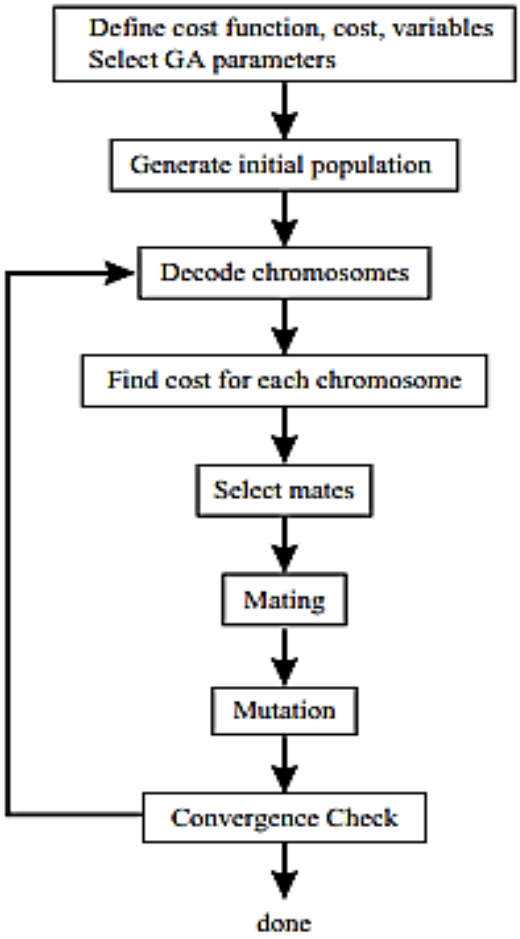

Figure 2. Genetic algorithm implementation.

Particle swarm optimization (PSO) is a technique derived from the coordinated movement of a flock of birds. It is a distributed behavioral algorithm that conducts multidimensional searches [17]. Particle Swarm Optimization (PSO) is categorized as a metaheuristic method due to its minimal or nonexistent assumptions on solutions. The technique operates by relocating particles (structures) throughout the search area via efficient algorithms that use the particles' position and velocity. Consequently, all individuals in the swarm can rapidly converge to the global position and a near-optimal geographical location by emulating the flock's behavior and their flight histories.

In practice, in Particle Swarm Optimization (PSO), the behavior of each individual structure is influenced by either the best local or the best global individual, facilitating its navigation through a hyperspace (i.e., Potential Energy Surface). A structure can be enhanced by a feedback mechanism, allowing it to learn from previous experiences to modify its flight speed and trajectory (search areas). This technology utilizes symmetry restrictions in structure formation, thereby diminishing the search space and augmenting structural variety. Specifically, during the searches, a fixed proportion of novel structures is incorporated at each generation to augment structural diversity. PSO employs structural characterization techniques to exclude analogous structures from the swarm. The program then utilizes local structural optimization to diminish the noise of the energy surfaces and guarantee the creation of physically valid structures [18].

2.3 Particle Swarm Optimization

Particle Swarm Optimization comprises two variants: global and local PSO algorithms, both of which have been effectively employed in several applications [19]. The global PSO exhibits rapid convergence, whereas the local PSO effectively mitigates premature convergence, hence improving its capacity to manage more intricate systems. CALYPSO, an acronym for Crystal structure AnaLYsis by Particle Swarm Optimization, serves as the primary implementation of this technology. This is an efficient method for structure prediction, accessible as a complimentary package for predicting and determining crystal structures and designing multifunctional materials [20]. This program is widely utilized and has been integrated with numerous local structural optimization codes (VASP, QE, GULP, SIESTA, CP2K, CASTEP), ranging from highly precise DFT methods to rapid semiempirical techniques capable of handling extensive systems. This PSO-based approach

[], in conjunction with fingerprint and matrix bond analysis, has effectively addressed many structural challenges [21], including the forecasting of novel high-pressure superconducting hydrides [22].

2.4 PSO Algorithm

A swarm of particles updates their relative positions from iteration to another, boosting the PSO algorithm to duly perform the search process. To get the optimum solution, each particle moves towards its prior personal best position (pbest) and the global best position (gbest) in the swarm [23]. Assuming a minimization problem, one have

$$\mathbf{p}_{best_i}^t = \mathbf{x}_i^* \mid f\left(\mathbf{x}_i^*\right) = \min_{k=1,2,\ldots,t} \left(\left\{f\left(\mathbf{x}_i^k\right)\right\}\right), \tag{1}$$

where $i \in \{1, 2, \ldots, N\}$, and

$$\mathbf{g}_{best}^t = \mathbf{x}_*^t \mid f\left(\mathbf{x}_*^t\right) = \min_{\substack{i=1,2,\ldots,N \\ k=1,2,\ldots,t}} \left(\left\{f\left(\mathbf{x}_i^k\right)\right\}\right), \tag{2}$$

where *i* denotes particle's index, *t* is the current iteration's number, *f* is the objective function to be optimized (minimized), x is the position vector (or a potential solution), and *N* is the total number of particles in the swarm. The following equations update, at each current iteration t+1, the velocity v and position x of of each particle *i* as:

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + c_1 \mathbf{r_1} \left(\mathbf{p}_{best_i}^t - \mathbf{x}_i^t\right) + c_2 \mathbf{r_2} \left(\mathbf{g}_{best}^t - \mathbf{x}_i^t\right), \tag{3}$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \tag{4}$$

where v represents the velocity vector, ω is the inertia weight utilized to balance the local exploitation and global exploration, r1 and r2 are random vectors uniformly distributed within the range [0,1]D (*D* being the search space dimensionality or the size of the problem at hand), and c1 and c2, called "acceleration coefficients", are positive constants.

An upper bound is commonly set for the velocity vector. As a means to prevent particles from shaving off the search space and forcing them to take a proper step size to comb the entire search domain, the "velocity clamping"

method was used [24]. The "constriction coefficient" strategy is another method, proposed by Clerc and Kennedy [25], in which the velocities can be also constricted by theoretically observing and analyzing the swarm dynamics.

3. The Hybrid GA-PSO used for Phishing Classification

This model combines both GA and PSO in a hybrid format. As can be seen in Figure 3.

1. **Initialization:**
   - Generate a population of solutions randomly or using heuristic methods.
   - Assign initial positions and velocities if starting with PSO.
2. **Optimization Loop:**
   - **Phase 1 (GA Exploration):**
     - Evaluate fitness of solutions.
     - Perform selection, crossover, and mutation.
     - Identify the best solutions.
   - **Phase 2 (PSO Exploitation):**
     - Update particle velocities and positions.
     - Use global and local best solutions for guiding the search.
     - Evaluate fitness and update best solutions.
3. **Convergence Check:**
   - Stop if the convergence criteria (e.g., maximum iterations, error threshold) are met.
   - Otherwise, alternate between GA and PSO or combine their operators dynamically.
4. **Output:**
   - The best solution found during the iterations.

5. Dataset

This dataset contains 48 features extracted from 5000 phishing webpages and 5000 legitimate webpages, which were downloaded from January to May 2015 and from May to June 2017. An improved feature extraction technique is employed by leveraging the browser automation framework (i.e., Selenium WebDriver), which is more precise and robust compared to the parsing approach based on regular expressions.

Anti-phishing researchers and experts may find this dataset useful for phishing features analysis, conducting rapid proof of concept experiments or benchmarking phishing classification models.

Figure 3. Visualization of dataset

6.The Constructed Model

The following represents the constructed model

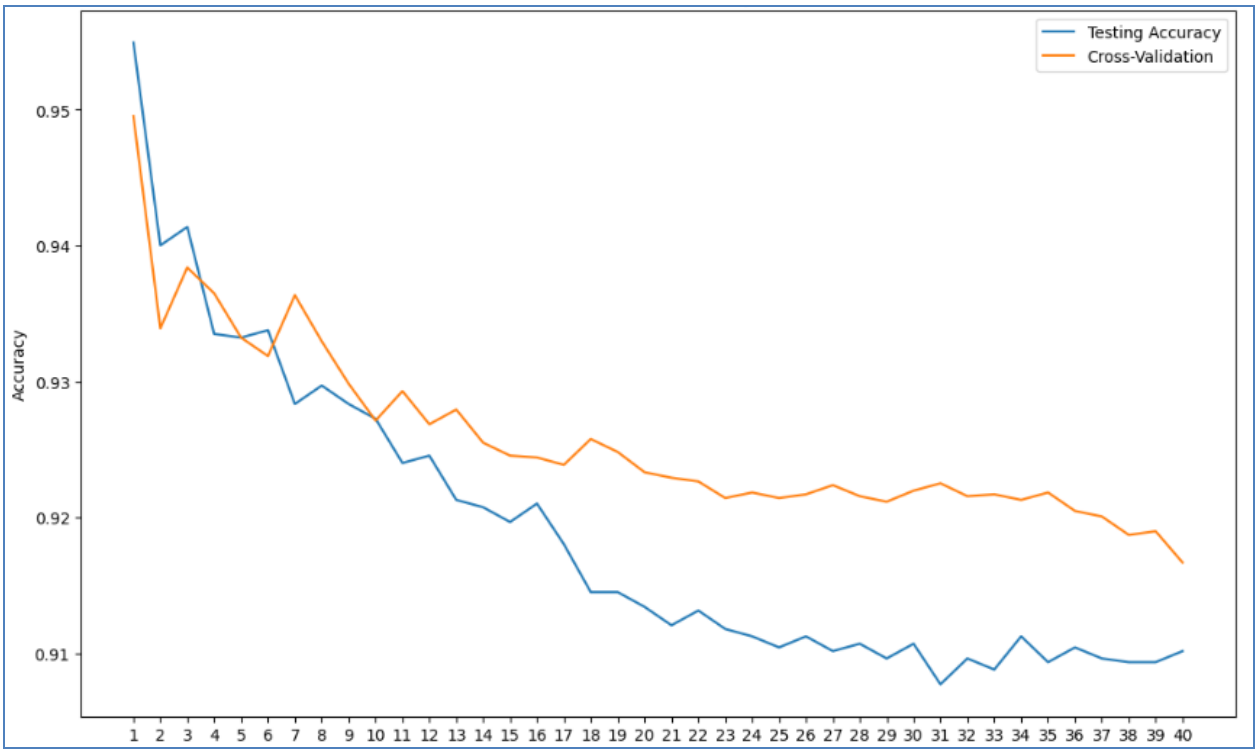

Figure 3: convergence of the proposed method

Table 1. output results after applying the proposed method GA-PSO

Predicted / True	Legitimate	Phishing	All
Legitimate	1951	151	2102
Phishing	180	1403	1583
All	2131	1554	3685

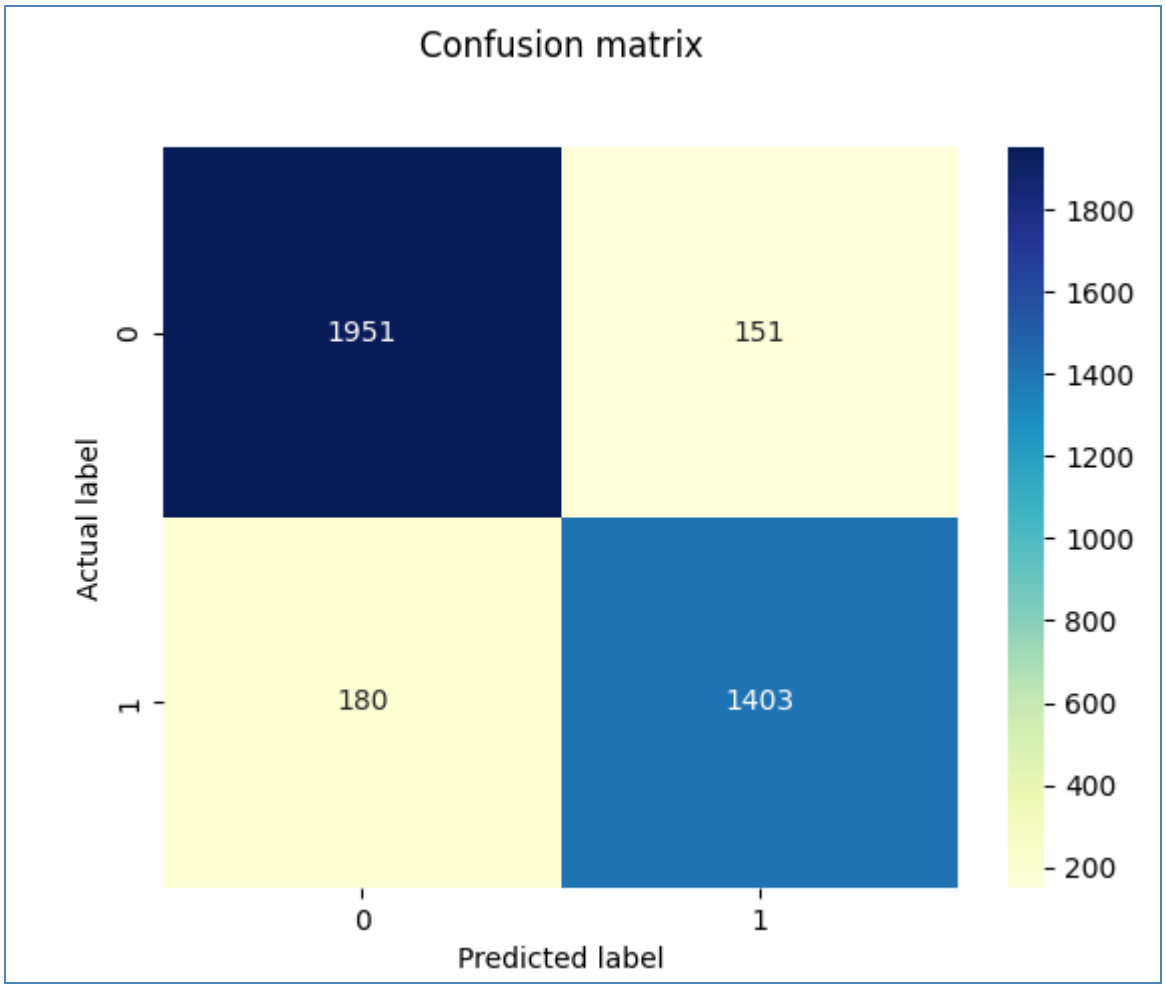

Figure 4. The actual vs. predicted classes in the confusion matrix

Table 2. output results of the precision, recall and f1-score of the proposed method

Precision	recall	f1-score	support
Legitimate	0.92	0.93	0.92	2102
Phishing	0.90	0.89	0.89	1583
Accuracy			0.91	3685
Macro avg	0.91	0.91	0.91	3685
Weighted avg	0.91	0.91	0.91	3685

7.Discussion and Conclusion

Particle Swarm Optimization (PSO) is a widely used meta-heuristic algorithm with notable strengths and weaknesses. While it has no formal proof of convergence, it is competitive in terms of convergence speed and accuracy compared to other methods, such as Evolutionary Algorithms (EAs). In the provided code, PSO is combined with a Genetic Algorithm (GA) to optimize feature selection on a phishing dataset. This hybrid approach leverages the strengths of both algorithms to mitigate their respective weaknesses.

PSO's main advantage lies in its ability to navigate high-dimensional search spaces efficiently, making it suitable for feature selection tasks. In the code, particles (solutions) represent binary vectors corresponding to selected features. PSO updates these particles iteratively, using social and cognitive factors to balance exploration and exploitation. The update_particle function dynamically adjusts particle positions and velocities based on the global best and personal best solutions, which enables rapid convergence to promising subsets of features.

However, PSO is not without its challenges. It is sensitive to hyperparameters, such as the inertia weight (w) and cognitive/social coefficients (c1 and c2), which require careful tuning to prevent premature convergence or divergence. Additionally, PSO's inherent focus on exploitation can lead to stagnation in local optima. To address these issues, the code integrates GA operations like crossover and mutation. GA enhances diversity in the population by introducing new feature subsets, counteracting PSO's tendency to converge prematurely.

The hybrid approach balances the exploration capabilities of GA with the convergence efficiency of PSO. While PSO refines solutions quickly, GA ensures the population does not stagnate by generating novel candidates through recombination and mutation. This synergy improves the algorithm's overall performance, making it robust for feature selection tasks in high-dimensional data.

In summary, the hybrid GA-PSO algorithm in the code exemplifies how PSO's fast convergence and adaptability can be combined with GA's diversity mechanisms to overcome challenges like premature convergence and limited exploration. This integration highlights the importance of balancing the strengths and weaknesses of meta-heuristic algorithms for practical optimization tasks.

8.References

1- Kocyigit, Emre, et al. "Enhanced feature selection using genetic algorithm for machine-learning-based phishing URL detection." Applied Sciences 14.14 (2024): 6081.

2- Alsenani, Theyab R., et al. "Intelligent feature selection model based on particle swarm optimization to detect phishing websites." Multimedia Tools and Applications 82.29 (2023): 44943-44975.

3- Anupam, Sagnik, and Arpan Kumar Kar. "Phishing website detection using support vector machines and nature-inspired optimization algorithms." Telecommunication Systems 76.1 (2021): 17-32.

4- Ali, Waleed, and Sharaf Malebary. "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection." IEEE Access 8 (2020): 116766-116780.

5- Alshahrani, Saeed M., et al. "URL Phishing Detection Using Particle Swarm Optimization and Data Mining." Computers, Materials & Continua 73.3 (2022).

6- Wang, Jiachen. "An improved genetic algorithm for web phishing detection feature selection." 2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML). IEEE, 2022.

7- Nordin, Noor Syahirah, et al. "A comparative analysis of metaheuristic algorithms in fuzzy modelling for phishing attack detection." Indones. J. Electr. Eng. Comput. Sci 23.2 (2021): 1146-1158.

8- Ghaleb, Sanaa AA, et al. "Feature selection by multiobjective optimization: Application to spam detection system by neural networks and grasshopper optimization algorithm." IEEE Access 10 (2022): 98475-98489.

9- Pathak, Prakash, and Akhilesh Kumar Shrivas. "Development of Proposed Model Using Random Forest with Optimization Technique for Classification of Phishing Website." SN Computer Science 5.8 (2024): 1-20.

10- Nagunwa, Thomas. "Comparative Analysis of Nature-Inspired Metaheuristic Techniques for Optimizing Phishing Website Detection." Analytics 3.3 (2024): 344-367.

11- Vidal, R., Ma, Y., and Sastry, S. (2005) discuss Generalized Principal Component Analysis (GPCA), a mathematical approach for analyzing and reducing data dimensionality, often used in pattern recognition and signal processing. This method could provide a theoretical foundation for feature selection.

12- Gupta, K. (2020) explains the utility of SelectKBest, a feature selection technique that evaluates the importance of features based on statistical tests, commonly used in machine learning workflows.
13- Li, F., Lai, L., and Cui, S. (2021) examine the robustness of LASSO (Least Absolute Shrinkage and Selection Operator), a popular regularization technique for feature selection under adversarial conditions.
14- Shobana, G., and Bushra, S.N. (2020) propose a tree-based feature selection model applied to classify myopia in children, demonstrating the practical application of feature selection in medical datasets.
15- Rey, C.C.T., García, V.S., and Villuendas-Rey, Y. (2023) explore evolutionary algorithms for feature selection in imbalanced datasets, relevant to cases where certain classes are underrepresented.
16- Catal, C., et al. (2022) provide a systematic review of how deep learning can be applied to detect phishing, a critical domain for cybersecurity.
17- Opara, C., et al. (2024) discuss a methodology for identifying phishing web pages by analyzing raw URL and HTML characteristics, combining feature extraction with machine learning.
18- Adebowale, M.A., et al. (2023) propose an intelligent detection framework using deep learning techniques to enhance phishing identification accuracy.
19- Shahrivari, V., et al. (2020) describe the use of machine learning models for phishing detection, with experiments detailed in the arXiv repository.
20- Venkatesh, B., and Anuradha, J. (2019) provide a comprehensive review of feature selection techniques, categorizing methods into filter, wrapper, and embedded approaches.
21- El-Hasnony, I.M., et al. (2020) propose a feature selection framework tailored for big data analytics, addressing the challenges of scalability and computational complexity.