Enhanced Phishing Website Categorization Using Random Forest with Sea Horse and Jellyfish Search Optimization

Yin Chen[1*], Yuan Ma[2]， Juan Zhao[2]， Yongqiang Zhang[2]
[1]Chongqing Institute of Engineering, Banan, Chongqing 400056, China
[2]Xi'an Siyuan University, Xi'an Shaanxi 710038, China

Keywords: categorization; phishing; personal information; random forest classifier; sea horse optimizer; jellyfish search optimization algorithm.

Received: January 18, 2025

In contemporary society, with advancements in science and technology, many global activities, ranging from financial transactions to information transfers, are conducted through the Internet via dedicated websites and applications. Unfortunately, the prevalence of online platforms has increased the proliferation of fake websites aimed at exploiting sensitive data, such as bank card information and personal details. It addresses the problem of cybersecurity w.r.t. the categorization of a set of 1353 websites by a machine learning algorithm into three categories, namely phishing, suspicious, and legitimate URLs. The dataset was gathered from published papers and divided into 70-30 in the training and testing phases. This will help keep members' banking and personal data much safer online. This paper uses the RFC model with two optimization schemes, Sea Horse Optimizer (SHO) and Jellyfish Search Optimization Algorithm (JSOA), to improve performance. After that, optimized versions of the schemes are tagged as RFSH and RFJS, respectively. After extensive training and testing on these three schemes, the best model was identified by comparing the performances of the three on the database in hand. The RFSH model performed better predicting, achieving 0.952 for all the data. It outperformed the RFJS model with a precision of 0.932 and the RFC single framework with an accuracy of 0.9106. Hence, it emerged as the best-predicting model.

Povzetek: Opisana je metoda za kategorizacijo spletnih strani kot lažnih, sumljivih ali legitimnih, ki uporablja klasifikator naključnih gozdov z metahevristično optimizacijo na osnovi morskega konjička in meduze; pristop izboljša učenje in zanesljivost brez ročnega prilagajanja.

1 Introduction

Phishing typically entails the creation of a fraudulent website with a sophisticated resemblance to a legitimate and trusted business site, designed to trick users and illicitly acquire their credentials, including login information [1]. The malevolent intent of phishers is to exploit acquired credentials to unlawfully access sensitive financial records—such as bank account numbers and credit card details—making the early detection and classification of malicious websites a critical step in preventing such breaches [2]. The significance of the Internet extends beyond member users to encompass organizations engaged in online business activities. Many enterprises provide online trading and services and goods sales [3]. Regrettably, the repercussions of falling victim to phishing are severe for users, as they become susceptible to identity theft and information breaches [4]. In a typical phishing attack, the initiation involves dispatching an email that seems to be from a lawful organization to potential victims. These emails prompt users to click on a URL embedded within it, encouraging recipients to modify their login information [5]. Alternative means of disseminating phishing Uniform Resource Locators (URLs) encompass Black Hat search engine optimization (Black Hat SEO) [6], peer-to-peer file sharing, blogs, forums [7], instant messaging (IM) [8], and Internet Relay Chat (IRC) [9].

Phishing toolkits, utilized by attackers, consist of compressed files containing replicated legitimate login pages (HTML, PHP, etc.) and associated resource files (images, CSS, logos, favicons, JavaScripts, etc.). These compressed files are then uploaded to a web server for hosting the phishing site, with attackers choosing various web servers, including compromised ones, free hosting providers, or paid services. The designers of these toolkits take precautions to evade traditional anti-phishing measures, employing techniques such as omitting or misspelling brand names found in titles, URLs, copyright information, headers, and descriptions. Additional tactics involve removing anchor links originally directed to legitimate URLs and replacing them with null anchor links. While attackers have the option to develop phishing sites from scratch, it is a resource-intensive process in terms of both time and design [10], [11], [12], [13], [14], [15], [16].

The categorization of phishing attacks is explained as follows [17]:

- *Deceptive Phishing:* Deceptive Phishing is a prevalent form of cyber-attack in which attackers impersonate legitimate companies to acquire personal information and login passwords illicitly.

Subsequently, they use this information to blackmail users to comply with their demands.

- *Spear Phishing:* Spear Phishing involves analyzing wireless network traffic using a wireless intrusion detection prevention system. While effectively detecting unauthorized wireless networks, it falls short in identifying suspicious activity at the application layer, transport layer, and protocol activities within a specific monitoring range.
- *Clone Phishing:* Clone Phishing entails creating an identical or cloned email using a legal or previously acquired email's attachment, link, and recipient address. The link is then substituted with a harmful version and transmitted to the victim from a spoofed email address, potentially leading to information compromise or gaining a foothold on another machine.
- *Whaling:* Whaling targets high-profile members, aiming to extract information through mediums like social media. Victims, called Whales or Big Phishers, are subjected to attacks similar to Spear Phishing, which involves acquiring personal data for malicious purposes.
- *Link Manipulation:* This is a phishing tactic in which a phisher sends a link to a spoofed or malicious website. When opened, the link directs the user to the phisher's website instead of the one mentioned. Still, users can prevent being deceived by manipulated links by checking the actual address before clicking.
- *Voice Phishing:* This, also known as vishing, is a phone-based violent crime that employs social engineering through the telephone system. It aims to extract personal and financial information for illicit economic activities.

Certain threat intelligence entities specializing in security identify and disclose malevolent web URLs or IPs, offering a blocklist database. This proactive approach aids in safeguarding others from the deleterious impacts of phishing.

Two distinct approaches are employed to differentiate between legitimate and phishing websites. The first method involves checking whether the requested URL is present on blocklists and comparing it with entries in those lists [18]. The second approach utilizes meta-heuristic tactics, wherein an extensive set of traits is retrieved from the site to categorize it as authentic or fraudulent [19]. The accuracy of the meta-heuristic tactic relies on retrieving a distinctive set of traits crucial for discerning between website types [20]. Data mining techniques are commonly employed to extract website traits to uncover patterns and relationships [21]. The significance of data mining schemes lies in their role in decision-making, as decisions can be informed by rules derived from these schemes [22].

Table 1 shows the literature review on literature reviews on studies that utilized various phishing website detection techniques.

Table 1: Literature reviews on studies that utilized various phishing website detection techniques.

No.	*Utilized approach*	*Algorithm*	*Dataset*	*Year*	*Ref.*
1	Heuristic and ML	Random Forest and Multilayer Perceptron	UCI ML Repository, 11,000 URL instances, and 30 features	2018	[23]
2	Heuristic and ML	Random Forest	PhishTank 11,055 instances and 30 features	2020	[24]
3	Visual Similarity, Heuristic and ML	Logistic Regression	PhishTank Yahoo URLBlacklist DMOZ	2019	[25]
4	Visual Similarity, Heuristic and ML	Support Vector Machine Random Forest Decision Tree K-Nearest Neighbor XGBoost Gradient Boosting and LightGBM	PhishTank Alexa 2000 web pages	2019	[26]
5	Visual Similarity and ML	Logistic Regression	OpenPhish PhishTank PhishStats	2021	[27]

6	Blacklist-Based, Visual Similarity, Heuristic, and ML	Adaptive Neuro-Fuzzy Inference System (ANFIS) Nave Bayes, PART, J48 Tree, and JRip	PhishTank MillerSmiles Relbanks	2021	[28]
7	List-Based, Visual Similarity, Heuristic, and ML	Support Vector Machine Random Forest Decision Tree AdaBoost XGBoost	PhishTank (4097 instances) and Google (5438 instances)	2020	[29]
8	ML and Heuristic	K-Nearest Neighbor, Logistic Regression, and Random Forest	Kaggle 11,504 URL with 32 attributes	2020	[30]
9	ML	Random Forest Decision tree	Kaggle 11,504 URL with 32 attributes	2020	[31]
10	ML and Heuristic	Support Vector Machine LightGBM Multilayer Perceptron Convolution Neural Network	ISCXURL-2016, 2978 instances, and 77 different features	2021	[32]
11	ML and Heuristic	Support Vector Machine, Grey Wolf Optimizer algorithm, Bat Algorithm, Whale Optimization Algorithm, Firefly Algorithm	PhishTank Yahoo UCI ML repository	2021	[33]
12	Deep Learning, Heuristic, and ML	Neural Network K-Nearest Neighbor Logistic Regression Support Vector Machine Gradient boosting, Ada-boost, and Random Forest	GitHub	2020	[34]
13	Deep Learning, Heuristic, and ML	Multilayer Perceptron Neural Network	Kaggle (10,000 web pages), ten features	2020	[35]
14	Convolutional Neural Network (CNN)	Long Short-Term Memory (LSTM)	651191 URLs	2024	[36]
15	ML and Deep Learning	XGBoost classifier, CNN, LSTM, and two hybrid models (CNN-LSTM and LSTM-CNN)	88647 instances	2024	[37]
16	Reinforcement Learning	Q-Learning-based	Large URLs dataset	2024	[38]
17	Reinforcement Learning	SmartiPhish	83275 instances	2024	[39]

The research on large-scale sophisticated machine learning with data mining methods will be highly accurate in differentiating between phishing, suspicious, and benign websites, lending considerable importance to increasing the user's information security. This work will further develop the effectiveness by adding two new

optimization schemes: SHO and JSOA. This cautious approach adds to cybersecurity strategy development and further convinces one to commit to improving and optimizing the prevalent framework to provide more accurate and reliable outcomes in detecting potential threats within the online environment.

2 Materials and methodology

2.1 RFC

RF is a type of supervised ML used for categorization and anticipation problems. In categorization problems, the performance of a random forest is excellent. A forest means multiple DTs and grows stronger with more trees. Every tree will be constructed using different data samples using the RFC method. Each of these trees predicts new data points independently and is involved in the voting system of the decision-making. The ultimate forecast (Cl_{rf}^{B}) is derived from most voting mechanisms, categorizing it as an ensemble tactic. This collective strategy, which utilizes uncorrelated tree schemes, outperforms a member model by mitigating errors and improving overall precision via the varied inputs contributed to the final forecast.

In developing DTs, trait retrieval and pruning techniques are vital. The Gini Index method [40] is particularly notable for trait retrieval in RFC, evaluating trait inconsistency concerning their classes. This method assesses inconsistency by haphazardly choosing a sample from the training set and predicting its class as Cl_i. The trait retrieval is expressed through the formula, where $\frac{F(Cl_i,T)}{(|T|)}$ displays the likelihood that a selected case belongs to Cl_i [41].

$$\sum\sum_{j\neq i}(F(\,Cl_i,T)/(|T|)(F(Cl_j,T)/(|T|) \tag{1}$$

Two critical parameters must be defined in constructing an RFC anticipation model: the count of trees (N, user-defined) and the input variables assigned to each tree. Comprising N DTs, RFC jointly utilizes their anticipations to determine the class of new data points through a voting mechanism [42].

2.2 Sea Horse optimizer (SHO)

Zhao et al. [43] introduced the SHO, a novel metaheuristic inspired by swarm intelligence and derived from the unique actions of SHs, including their transience, hunting strategies, and birthing strategies. The SHO is designed to adapt and survive in its environment, drawing inspiration from these key characteristics of SHs.

SHO entails four steps: 1_initialization, 2_mobility, 3_predation, and 4_breeding, with thorough explanations below.

2.2.1 Initialization step

Like various other metaheuristic schemes, SHO initiates by establishing the group. In this instance, the SHs within the group are potential solutions to an issue within the search domain, expressed through Eq. (2):

$$S=\begin{bmatrix} x_1^1 & \ldots & x_1^d \\ \ldots & \ldots & \ldots \\ x_P^1 & \ldots & x_P^d \end{bmatrix} \tag{2}$$

d displays the dimensionality of the variable, P signifies the group volume, and s displays the SHs in the group.

To create member solutions, the upper bound (Ub) and lower bound (Lb) of the problem were deployed as first spots for random creation. The process for creating the $i-th$ member, X_i, in the search domain [Lb, Ub], is outlined by Eqs. (3) and (4).

$$X_i=\left[x_i^1,\ldots,x_i^d\right] \tag{3}$$

$$x_i^j = rand*(Ub^j-Lb^j)+LB^j \tag{4}$$

Ub and Lb for the $\mathrm{j}-\mathrm{th}$ variable in the enhanced issue is displayed as Ub^{j} and Lb^{j}, accordingly.

rand displays a random number within the [0, 1] range, where:

j displays an integer from 1 to d (dimensionality of the problem), d displays the dimensionality of the problem, i displays a positive integer from 1 to P (size of the group), P displays the size of the group, $\mathrm{x_i^j}$ signifies the $\mathrm{j}-\mathrm{th}$ aspect of the $\mathrm{i}-\mathrm{th}$ member in the group.

When addressing a minimum/maximum enhancement issue, the member exhibiting the lowest/highest fitness degree is recognized as $\mathrm{X_b}$, showcasing the best resolution. The value of $\mathrm{X_b}$ can be calculated using Eq. (5):

$$X_b=\arg_{\min or\max}\left(f(X_i)\right) \tag{5}$$

$f(X_i)$ displays the value of the objective function for a particular task.

2.2.2 Transition step

SHs exhibit varied transition schemes influenced by a normal distributed random spread (0,1). Balancing exploration and exploitation, a cut-off point at $r_1=0$ divides SHs into halves for local and global search. Later algorithm steps handle the transition treatment.

First step:

SHs spiral in response to ocean vortices. If the chance value r_1 surpasses the SHO limit, the scheme emphasizes local exploitation. SHs move spirally toward the best resolution X_b, using Lévy flights for the step size. This benefits exploration in early cycles and prevents over-localization. The spiral transition dynamically adjusts the rotation angle, expanding the search domain. Eq. (6) is employed to create a fresh position for SHs.

$$\begin{aligned} X_{new}^1(t+1) &= X_i(t)+Levy\,(\lambda)((X_b(t) \\ &\quad -X_i(t)*x*y*z \\ &\quad +X_b(t)) \end{aligned}$$
$$h.t\begin{Bmatrix} x=p*\cos(\theta) \\ y=p*\sin(\theta) \\ z=p*\theta \\ p=u*e^{\theta v} \end{Bmatrix} \tag{6}$$

The parameters u and v characterize the logarithmic spiral, influencing the stem length (p), with a constant set to 0.05 for every u and v combination. The three-dimensional coordinates under spiral transition are displayed by $x, y,$ and z. θ is chosen haphazardly within the range of $[0, 2\pi]$.

The Lévy flight spreadfunction (Levy(z)) is gauged by Eq. (7).

$$Levy(z) = h * \frac{\omega * \sigma}{|k|^{\frac{1}{\lambda}}} \tag{7}$$

Random positive numbers w and k are chosen from the range of zero to one. The variable h is fixed at 0.01, and λ is haphazardly picked from the interval [0, 2], with a specific value of 1.5. The computation of σ is identified by applying Eq. (8).

$$\sigma = \left(\frac{\Gamma(1+\lambda) * \sin\left(\frac{\pi\lambda}{2}\right)}{\Gamma\left(\frac{1+\lambda}{2}\right) * \lambda * 2^{\left(\frac{\lambda-1}{2}\right)}}\right) \tag{8}$$

Second step:

This step illustrates SHs' Brownian transition, which was influenced by ocean waves. If r_1 is on the left of the restriction, SHO shifts to a drifting mode to avoid local optima. As described by Eq. (9), Brownian transition extends SHs' transition range for enhanced exploration in the search domain.

$$X_{new}^1(t+1) = X_i(t) + rand * l * \beta_t * (X_i(t) - \beta_i * X_{best}) \tag{9}$$

$$h.t\left\{\beta_t = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)\right\}$$

The random walk coefficient for Brownian transition is displayed by β_i, and a fixed parameter with a value of 0.05 is represented by the symbol l.

The fresh position of the SH at cycle t can be gauged by mixing the two described situations with the use of Eq. (10):

$$X_{new}^1(t+1) = \begin{cases} X_i(t) + Levy(\lambda)\left(\left(X_b(t) - X_i(t)\right) * x * y * z + X_b(t\right. \\ X_i(t) + rand * l * \beta_t * (X_i(t) - \beta_i * X_b), \ r_1 : \end{cases} \tag{10}$$

2.2.3 Predation step

When SHs hunt for zooplankton, success or failure is identified by a random number, r_2, created by SHO. With a probability of over 90% for successful predation, the critical value of r_2 is set at 0.1. Effective hunting, indicated by r_2 greater than 0.1, involves the SH approaching, overtaking, and capturing the prey (ideal resolution). Unsuccessful attempts result in sea horse and prey moving in the opposite direction, indicating continued exploration. The predation behavior is mathematically represented by Eq. (11).

$$X_{new}^2(t+1) = \begin{cases} \alpha * (X_b - rand * X_{new}^1(t) + (1-\alpha) * X_b, & if \ \ r_2 > 0.1 \\ (1-\alpha) * (X_{new}^1(t) - rand * X_b) + \alpha * X_{new}^1(t), & if \ \ r_2 \le 0.1 \end{cases} \tag{11}$$

The refreshed position of the SH after transition at cycle t is displayed by $X_{new}^1(t)$. The variable r_2 is a haphazardly created integer in the interval of 0 to 1.

The computation outlined in Eq. (12) determines that the SH's transition step size diminishes linearly with cycles when chasing prey.

$$\alpha = \left(1 - \frac{t}{T}\right)^{\frac{2t}{T}} \tag{12}$$

T displays the algorithm's maximum number of cycles.

2.2.4 Breeding behavior phase

The group is split into male and female groups based on fitness levels to address male SHs' breeding action. In SHO, members with the highest fitness become selected fathers, while the rest form the group of chosen mothers. Eq. (13) demonstrates that this separation hinders excessive localization of fresh approaches and facilitates the inheritance of beneficial traits by both mothers (M) and fathers (F), ultimately benefiting the next generation.

$$\begin{cases} F = X_{sort}^2(1:\frac{P}{2}) \\ M = X_{sort}^2(\frac{P}{2}+1:p) \end{cases} \tag{13}$$

X_{sort}^2 signifies the solutions X_{new}^2 sorted by increasing fitness values. In SHO, mothers and fathers match the female and male groups. It functions on the presumption that fresh offspring arise from the chance pairing of females and males. Efficiency is maintained by assuming each sea horse pair produces only one offspring, as illustrated in Eq. (14).

$$X_i^{offspring} = r_3 X_i^F + (1 - r_3) X_i^M \tag{14}$$

The variable i is a positive value in the range $[1, \frac{p}{2}]$, where p is a metric. X_i^F and X_i^M denote the haphazardly selected male and female members, respectively. The integer r_3 is haphazardly created and falls within the range [0, 1]. Fig. 1 displays the diagram of SHO. In this figure, t represents the iteration.

Figure 1: Flowchart of SHO

2.3 Jellyfish search algorithm (JSO)

A recent addition to swarm-based metaheuristics is JSO, developed by Chou and Truong in 2021 [44]. JSO mimics the way jellyfish search for food in the ocean [45].

2.3.1 Numerical framework

JSO adheres to three theoretical standards:

2.3.1.1. Marine flow

Jellyfish can sense the path of marine flows (as per Eq. (15)) to discover and consume smaller planktonic forms.

$$\overrightarrow{OC} = X' - \beta \times M \times d(0,1) \quad (15)$$

Here, $\vec{D}$ displays the direction of the marine flow, β ($\beta > 0$) is the coefficient defining the length distribution of $\overrightarrow{OC}$, X' displays the place of the current best jellyfish in the swarm, and M is the mean location of all jellyfish.

The refreshed position of every jellyfish can be articulated as follows:

$$X_i(t+1) = X_i(t) + d(0,1) \times \overrightarrow{OC} \quad (16)$$

After adjusting each jellyfish's situation, a favorable place, potentially with increased food source availability, is selected as the jellyfish's current position.

2.3.1.2. Jellyfish bloom

Jellyfish within a bloom display two kinds of transitions: passive and active. Here, mathematical schemes for these transitions are presented:

$$\text{Passive transition:} \quad X_i(t+1) = X_i(t) + \lambda \times d(0,1) \times (w_b - L_b) \quad (17)$$

Here, λ (λ > 0) is a coefficient linked to the degree of passive transition. The lower bound, w_b, and the upper bound, L_b, define the search domain.

$$\text{Active transition:} \quad X_i(t+1) = X_i(t) + r(0,1) \times \vec{D}$$

Were

$$\vec{D} = \begin{cases} X_i(t) - X_j(t) & if \quad g(X_i) < g(X_j) \\ X_j(t) - X_i(t) & if \quad g(X_i) \geq g(X_j) \end{cases} \quad (19)$$

The values of the objective function for jellyfish i and j are displayed as $g(X_i)$ and $g(X_j)$ respectively.

2.3.1.3. Time management system

It governs both the transition of the two types of jellyfish in the bloom and their transitions toward marine flows. The function representing this time control is expressed as:

$$T(t) = \left| \left(1 - \frac{t}{MaxIter}\right) \times (2 \times d(0,1) - 1) \right| \quad (20)$$

Here, t displays the time index, expressed as the cycle count, and $MaxIter$ displays the peak count of cycles.

2.3.2 Group initialization

The initial group is created using the Logistic map.

$$X_{i+1} = \upsilon X_i(1 - X_i), \quad 0 \le X_0 \le 1 \tag{21}$$

Here, X_i displays the chaotic value corresponding to the location of the $i-th$ jellyfish and X_0 displays a location created haphazardly. Throughout all experiments, the parameter υ is fixed at a value of 4.

2.3.3 Boundary regulation system

If a jellyfish exceeds the boundaries of the defined search domain, it will be positioned in those boundaries using Eq. (37).

$$\begin{cases} X'_{i,r} = (X_{i,r} - W_{b,r}) + L_{b,r} & if \quad X_{i,r} > W_{b,r} \\ X'_{i,r} = (X_{i,r} - L_{b,r}) + W_{b,r} & if \quad X_{i,r} < W_{b,r} \end{cases} \tag{22}$$

Here, $X_{i,r}$ and $X'_{i,r}$ display the present and refreshed place of the d-th dimension for the i-th jellyfish. $W_{b,r} and\ L_{b,r}$ denote the lower and upper bounds, respectively, for the $r-th$ dimension in the search domain.

The diagram for JSO is illustrated in Fig. 2, and the corresponding pseudocode is provided below.

Initialization phase
Define the parameters for the algorithm: number of population (n_{pop}) and maximum cycle ($MaxIter$) generate the primary group $X_i(i = 1,2, \dots, n_{pop})$ using Eq. (21)
Assess the initial group, $f(X_i)(i = 1,2, \dots, n_{pop})$
Identify the current top-performing jellyfish (X')
Initialize time: $t = 1$
Cyclic body of the algorithm
While $t \le MaxIter$
For $i = 1$: n_{pop}
Gauge the time regulation $T(t)$ using Eq. (20)
If $T(t) \ge 0.5$
Calculate the path of the marine flow using Eq. (15)
Compute the refreshed position of the $i-th$ jellyfish using Eq. (16)
Else
If $(1 - T(t)) < rand(0,1)$
Calculate the refreshed position of the $i-th$ jellyfish using Eq. (17)
Else
Compute the refreshed position of the $i-th$ jellyfish using Formulas (18) and (19)
End if
End if
Verify restriction situations using Eq. (22)
Examine the fresh place of the i-th jellyfish
Refresh the place of the i-th jellyfish (X_i)
Update the place of the currently best jellyfish (X')
End for i
$t = t + 1$
End while
Output the ideal outcomes

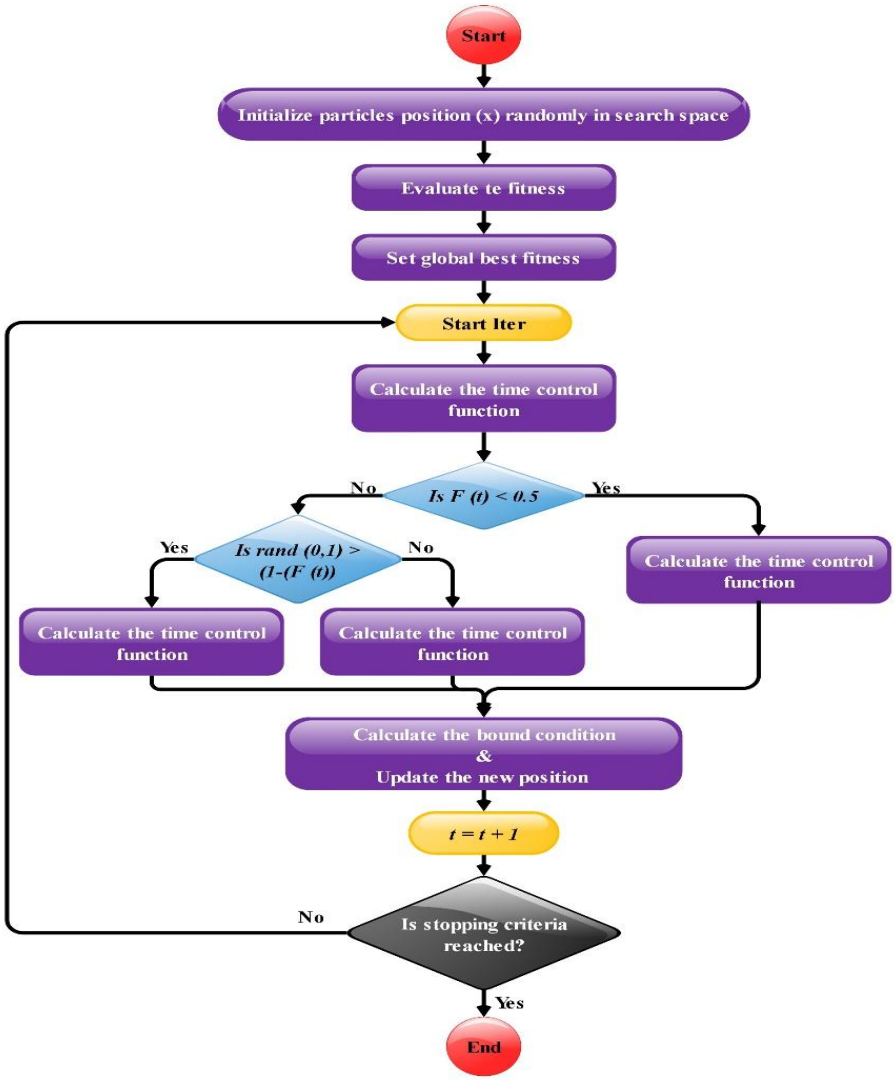

Figure 2: Flowchart of JSO

2.4 Data collection

An integral phase in the data mining procedure entails thoroughly preprocessing the database. During this step, textual information is meticulously transformed into numerical values, laying the basis for utilizing machine learning schemes and sophisticated statistical approaches. This is a critical transformation that will enable an in-depth analysis of the database and, therefore, yield material insight into its usage.

The database contains different factors organized carefully to distinguish between phishing websites and real ones. The data is presented in a structured framework that allows for in-depth, organized scrutiny for more accurate and reliable analysis in the future.

The present work examines a manifold of inputs over various dimensions that drastically affect the decision between phishing and legitimate websites. Its comprehensiveness embraces a wide range of elements, from technical indications to behaviorist trends, to delve deeply into the interactions related to web security. The dataset gathered from the published study [46].

- *SFH (Server Form Handler)*: Evaluates the security of form submissions on a webpage.
- *Pop-Up Window*: Assesses the presence or behavior of pop-up windows on a webpage.
- *SSL final state*: displays the final state of an SSL (Secure Sockets Layer) connection, providing insights into the website's security.
- *Request URL*: This examines the URLs requested during a web page's loading. It is often used to assess potential security threats.
- *URL of Anchor*: Analyzes the anchor (hyperlink) tags in the webpage, assessing the quality and security of linked URLs.
- *Web traffic*: Measures a website's web traffic or popularity, which can indicate its trustworthiness.
- *URL Length*: Assesses the length of the URL, as excessively long URLs may be associated with phishing or deceptive websites.
- *Age of domain*: Displays the age of the domain, with older domains often considered more trustworthy.

- *Having IP Address*: Showcases whether the URL has an IP address, and the presence of an IP address in the URL may be a security concern.

A diversified strategy like this would ensure completeness in the analysis regarding various factors that, when combined, help differentiate phishing websites from real ones. In this respect, it involves a broad appraisal of technical criteria, behavioral trends, and contextual influences. This is multi-dimensional research in this direction, aspiring to encapsulate the intricacy of detecting deceitful online actions versus real sites.

Fig. 3 presents the correlation matrix, illustrating the relationships between the input features and the target classification outcome (i.e., whether a URL is phishing, suspicious, or legitimate). The figure uses color intensity to represent the strength of the correlation, with darker shades indicating stronger relationships. The most influential feature in this analysis is "Having an IP Address," which shows the highest positive correlation score of 3.8, highlighting its critical role in distinguishing fraudulent websites. This aligns with cybersecurity principles, as phishing sites often use direct IP addresses to bypass domain reputation checks. Additionally, the SSL final state exhibits a moderately strong correlation of 1.9, indicating that the presence or absence of a secure SSL certificate substantially affects phishing likelihood. Another significant input is the domain age, which shows a positive correlation, emphasizing that newer domains are often associated with phishing behavior. Conversely, SFH presents a negative correlation of -0.70, suggesting that poorly configured or insecure form handlers are strongly associated with fraudulent websites. Overall, the correlation analysis underscores that most features negatively influence a website's authenticity, with only a few—like Web Traffic (0.21) and Having an IP Address (1.7)—exerting clear positive impacts.

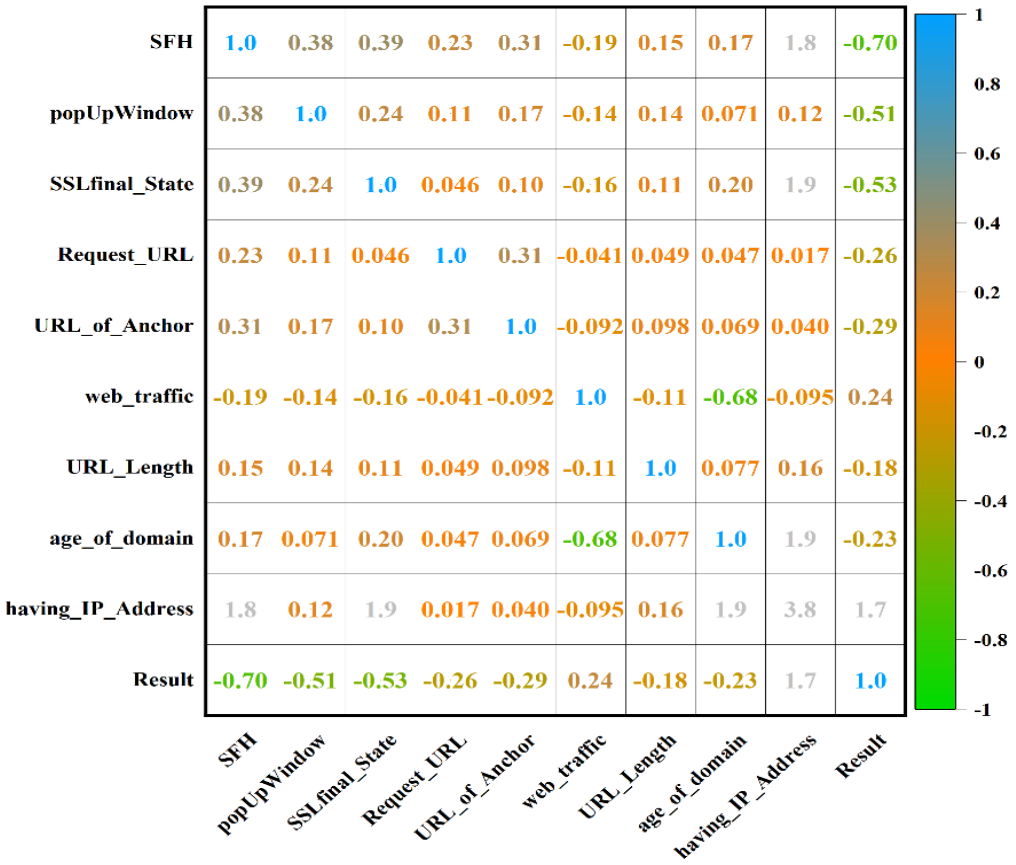

Figure 3: Correlation matrix for the link between inputs and output.

3 Outcomes

3.1 Framework practicality appraisal

Accuracy is a common metric for assessing a framework's total productivity in classifying issues. It relies on True Positives (T_P), True Negatives (T_N), False Positives (F_P), and False Negatives (F_N). However, Accuracy may be less effective in imbalanced data, favoring the majority class. To address this, Recall, Precision, and F1-Score are additional metrics offering nuanced insights into model performance, particularly in imbalanced situations. These metrics, expressed in numerical formulas (typically numbered 23 to 26), collectively refine the appraisal of a categorization model's effectiveness. In addition, Table 2 displays the formula of appraisal metrics.

Table 2: Formulation of appraisal metrics

Metrics	Number
$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$	(23)
$Precision = \frac{T_P}{T_P + F_P}$	(24)
$Recall = T_P R = \frac{T_P}{P} = \frac{T_P}{T_P + F_N}$	(25)
$F1\_score = \frac{2 \times Recall \times Precision}{Recall + Precision}$	(26)

3.2 Convergence outcomes

Convergence diagrams are widely used in scholarly communication to visually illustrate the optimization progress in schemes or schemes across cycles. Fig. 4 illustrates the convergence behavior of the two optimization-based models—RFSH and RFJS—over 200 learning cycles. Initially, both models start from similar baselines. The RFJS model demonstrates an early surge in performance during the first few cycles, indicating a faster initial convergence. However, this gain plateaus over time. In contrast, the RFSH model shows a more consistent and gradual improvement in accuracy throughout the training process. By the 200th cycle, RFSH achieves a superior final accuracy of approximately 0.91, compared to 0.89 for RFJS. This convergence pattern highlights the resilience and adaptive efficiency of the RFSH model, particularly in later training stages. The performance advantage of RFSH may be attributed to the exploitation–exploration balance managed by the Sea Horse Optimizer, enabling more refined learning during prolonged iterations. Both models were fine-tuned using a random search hyperparameter optimization technique, which explored multiple combinations of parameters to maximize model performance.

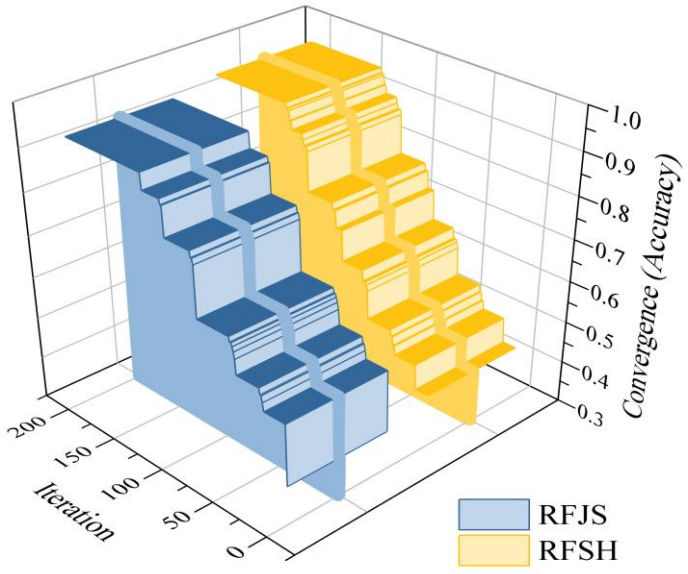

Figure 4: 3D ribbons for the convergence of hybrid models

3.3 Comparing outcomes of predictive schemes

Table 3 presents a thorough overview of the outcomes derived from the developed RFC schemes, providing a detailed insight into their performances. In addition, Fig. 5 employs a 3D bar chart to visually illustrate how assessors evaluated these schemes, aiming to pinpoint the model that excels in precision when predicting practical results. A considerable portion of the database endures intensive training, while the remaining values undergo meticulous testing. The outcomes, covering the entire database, are carefully noted in Fig. 5, graphically showcasing the essential parameter for scheme appraisal. Accuracy appraisal is performed in three phases: Train, Test, and All. The outcomes observed during the training phase across all three schemes expose superior performance relative to the subsequent phases. A comparative analysis of the data from both the training and test phases, as presented in the consolidated All phase, highlighted that the RFSH model, exhibiting an approximate value of 0.952, surpasses the optimized RFJS model with a value of 0.932 and the single RFC model, which records a value of 0.91. This highlights the RFSH model's distinction as superior in this evaluative context. The elucidated values from Table 3 are graphically depicted in Fig. 5. Notably, the RFC and RFJS schemes exhibit relatively consistent columnar patterns. However,

the RFHS model displays a decrement in performance during the test section, followed by a subsequent recovery in the all-encompassing section. This resurgence underscores its superiority and establishes a notable advantage over the other schemes.

Table 3: Result of RFC-based developed models in the training, testing, and All sections.

Phase	Index values	Schemes		
		RFC	RFJS	RFSH
Train	Accuracy	0.9113	0.9324	0.9567
	Precision	0.9104	0.9324	0.9570
	Recall	0.9113	0.9324	0.9567
	F1 _score	0.9102	0.9324	0.9567
Test	Accuracy	0.9089	0.9310	0.9409
	Precision	0.9095	0.9323	0.9423
	Recall	0.9089	0.9310	0.9409
	F1 _score	0.9083	0.9312	0.9409
All	Accuracy	0.9106	0.9320	0.9520
	Precision	0.9097	0.9320	0.9524
	Recall	0.9106	0.9320	0.9520
	F1 _score	0.9095	0.9320	0.9520

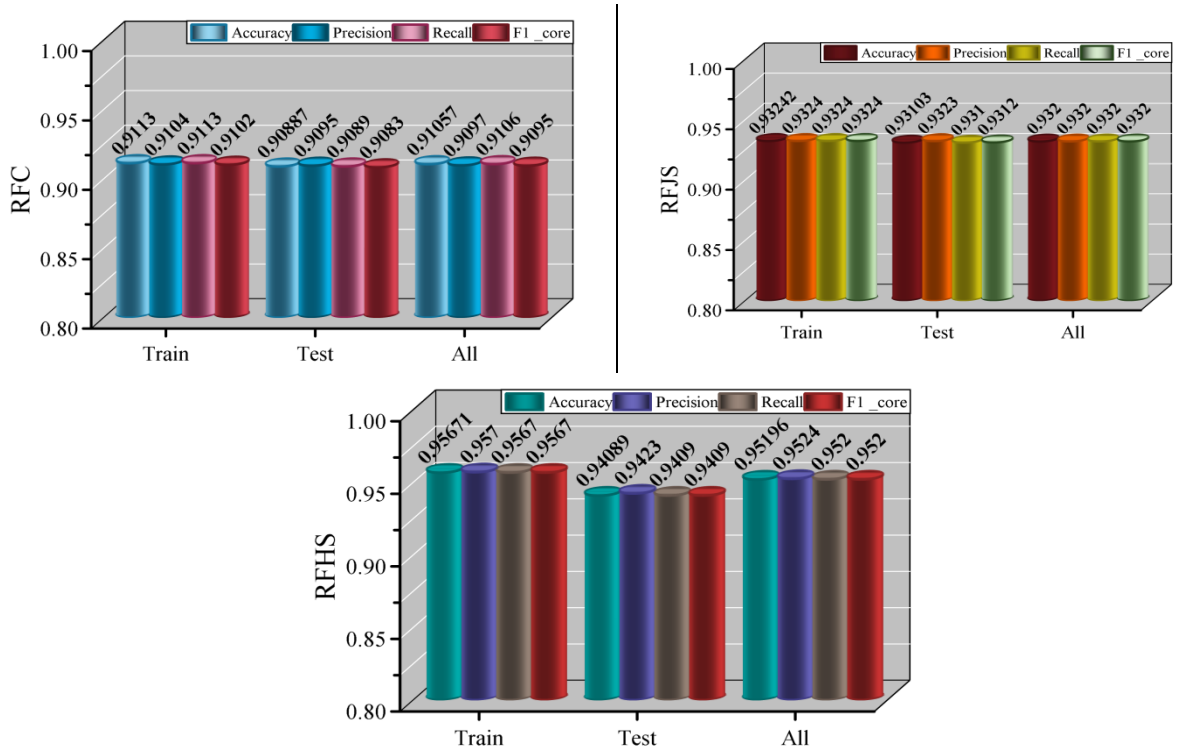

Figure 5: Achievement-based 3D bar chart for the developed models' productivity by evaluators

3.4 Categorization outcomes

Table 4 and Fig. 6. facilitate a comparison of the model's accuracy in predicting legitimate, phishing, and suspicious instances. Examining Fig. 6. indicates specific gauged values, where phishing instances are recorded as 548, suspicious instances as 702, and legitimate instances as 103. Notably, the RFSH model exhibits the highest correct anticipation rates across the three categories, with numerical values corresponding to 101 legitimate cases, 514 instances of phishing, and 673 instances of suspicious categories. Additionally, Table 4 underscores the precision values, revealing a remarkable match of 95.72% for phishing, 95.73% for suspicious, and 89.38% for legitimate instances within the RFSH model. This

underscores its pronounced superiority in accuracy. Further substantiating this superiority, the F1-score and Recall sections also reflect favorable outcomes for the RFSH model.

Table 4: Appraisal indexes of the designed schemes' productivity drawing on grades.

Model	Grade	Index values		
		Precision	Recall	F1-score
RFC	Phishy	0.918	0.9051	0.9084
	Suspicious	0.917	0.9444	0.9305
	legitimate	0.8488	0.7087	0.7725
RFJS	Phishy	0.9336	0.9234	0.9284
	Suspicious	0.9392	0.9459	0.9425
	legitimate	0.875	0.8835	0.8792
RFSH	Phishy	0.9572	0.938	0.9475
	Suspicious	0.9573	0.9587	0.958
	legitimate	0.8938	0.9806	0.9352

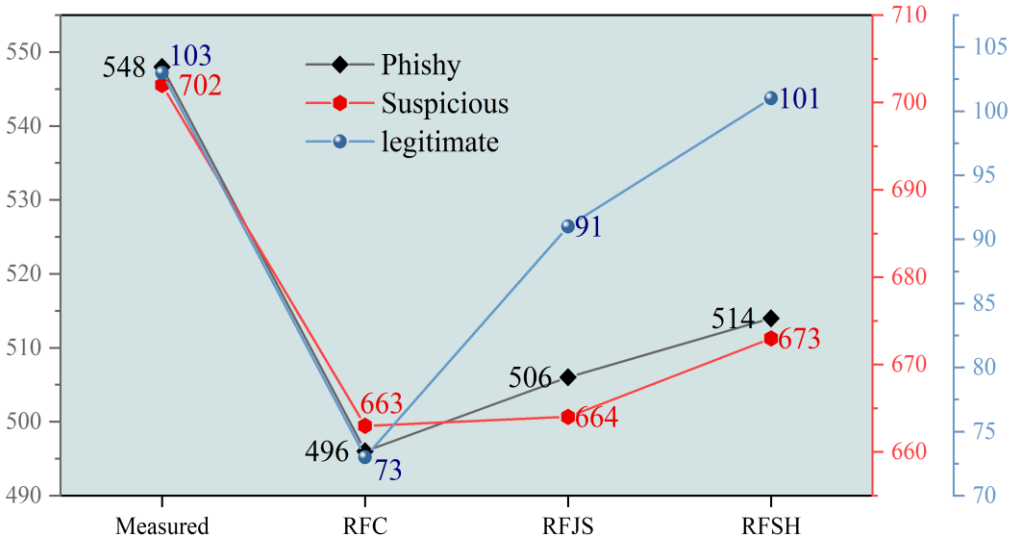

Figure 6: Line-symbol plot for comparing the measured and predicted values

Utilizing the confusion matrix enables the determination of miscategorizations for each model out of 1353 websites in Fig. 7. A detailed examination of the columns illustrates specific miscategorization numbers for each model. The RFC model exhibits 52 miscategorizations in phishy, 39 miscategorizations in suspicious, and 30 miscategorizations in legitimate instances. Similarly, the RFJS model records 42 miscategorizations in phishy, 38 miscategorizations in suspicious, and 12 miscategorizations in legitimate cases. On the other hand, the RFSH model manifests 34 miscategorizations in phishy, 29 miscategorizations in suspicious, and two miscategorizations in legitimate cases.

Given the fewest anticipation errors compared to the gauged values, RFSH rises as the most favorable scheme for future anticipations. This miscategorization reduction underscores its potential for enhanced accuracy and reliability in subsequent predictive scenarios.

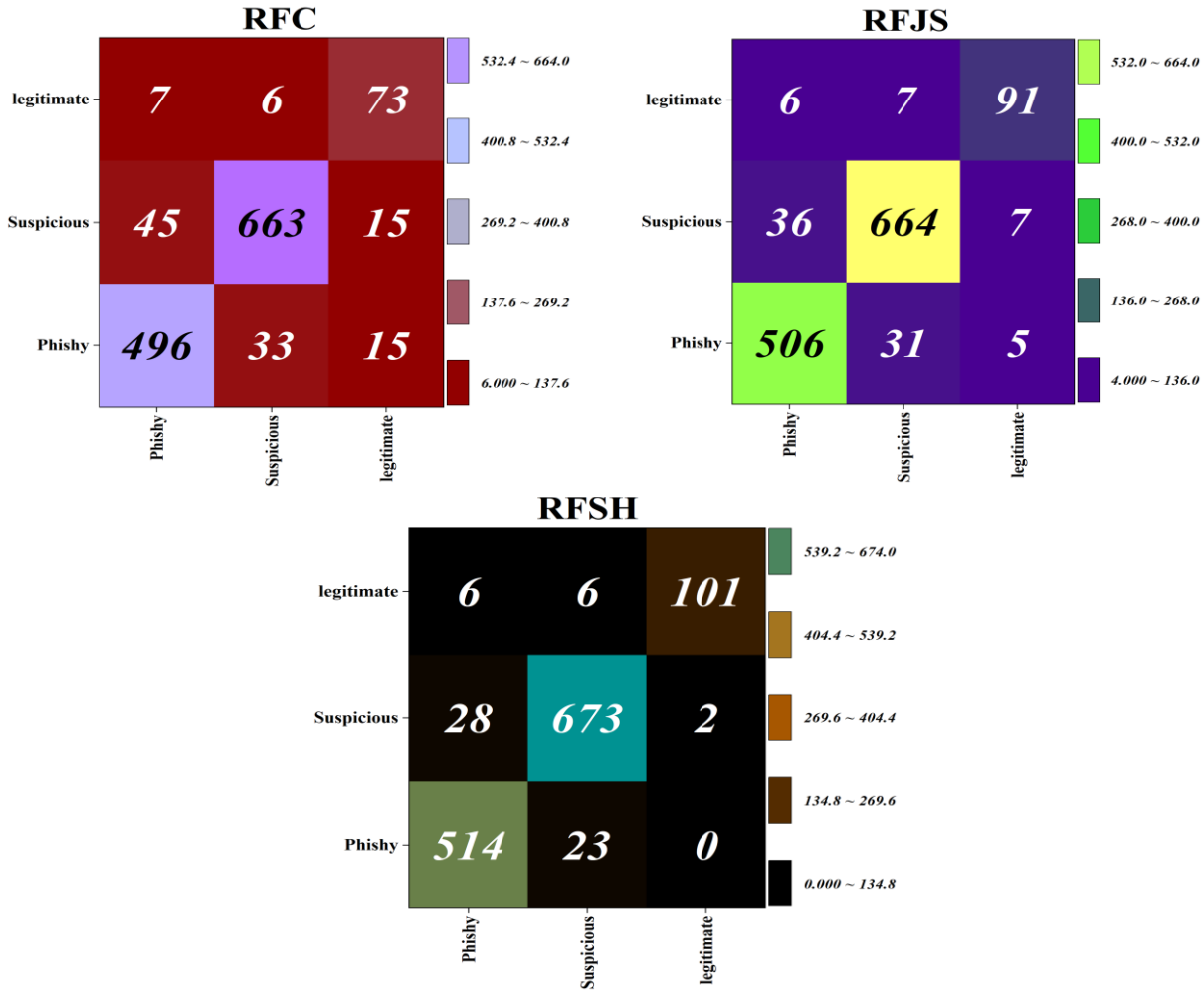

Figure 7: Confusion pattern showcasing the accuracy of each model

Fig. 8 presents the Receiver Operating Characteristic (ROC) curves for three models—RFC, RFJS, and RFSH—to evaluate their capability in distinguishing phishing, suspicious, and legitimate websites. The ROC curve illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) across different classification thresholds, where a curve closer to the top-left corner signifies superior classification performance. The baseline RFC model performs adequately with an Area Under the Curve (AUC) of 0.8993 but shows a comparatively higher FPR. Upon integrating optimization strategies, the RFJS model achieves a notable improvement with an AUC of 0.9385, reflecting enhanced sensitivity and reduced misclassifications. The RFSH model, with the highest AUC of 0.9654, demonstrates exceptional discriminative ability, with a steep curve that indicates high sensitivity and minimal false positives. This figure underscores the effectiveness of incorporating metaheuristic optimization algorithms into the Random Forest framework, resulting in significant performance gains, with the RFSH model emerging as the most robust and reliable for phishing detection.

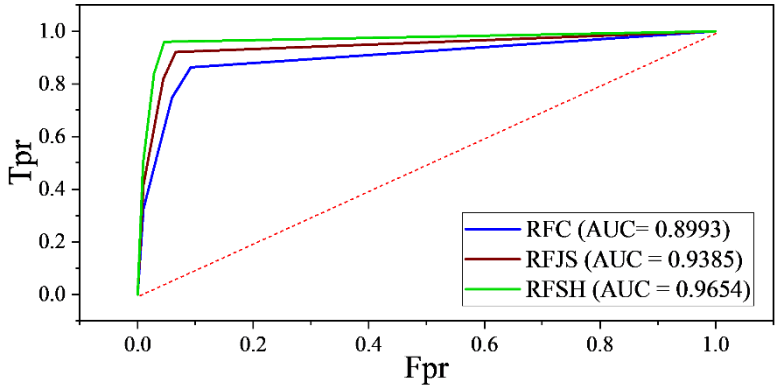

Figure 8: Receiver Operating Characteristic (ROC) curve for the top-performing hybrid model

The SHAP analysis results in Fig. 9 highlight the most influential features in categorizing websites as phishy, suspicious, or legitimate. Among all features, the SFH demonstrates the highest importance, particularly for identifying phishy and legitimate websites, indicating that insecure form handling is a strong indicator of phishing behavior. Pop-Up Windows and SSL Final State are impactful features, showing that deceptive pop-ups and lack of proper SSL implementation are common traits in phishing websites, while secure SSL practices are associated with legitimate sites. Request URL and URL of Anchor have moderate importance, emphasizing the role of embedded links in signaling potential malicious intent. Web traffic, URL length, and age of the domain offer additional discriminative power, especially in differentiating legitimate sites from suspicious or phishy ones. Notably, the feature of Having an IP Address has minimal influence across all categories, suggesting that its standalone effect is limited in modern phishing detection. Overall, this analysis affirms that the model relies on behaviorally relevant features and offers a transparent interpretation of how each input contributes to the classification outcome.

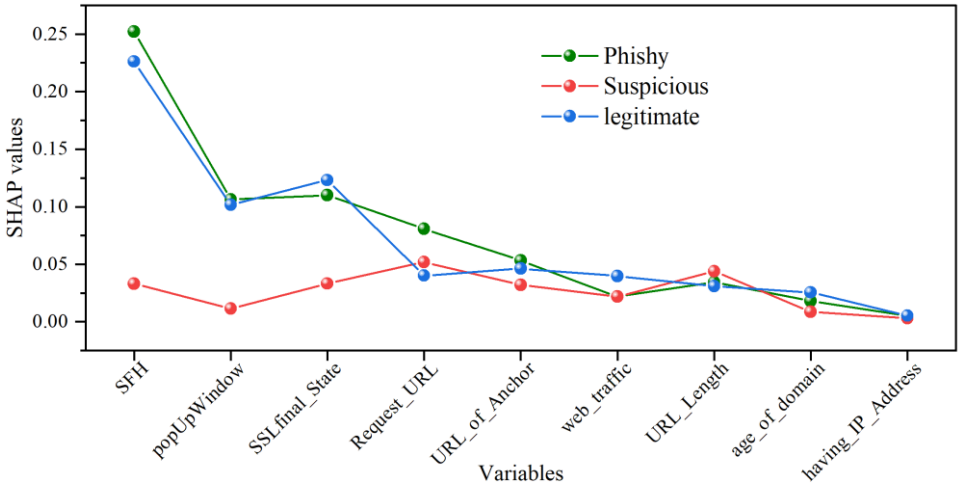

Figure9: Impact of the input variables on models' output based on SHAP sensitivity analyses.

To assess the statistical significance of the differences in the performance of the three models, including RFC, RFSH, and RFJS, a one-way Analysis of Variance (ANOVA) test was conducted. The results are presented in Table 5. The F-values for RFC (0.2523), RFSH (0.0587), and RFJS (0.0491) are all notably low, and their corresponding P-values are substantially greater than the conventional significance level of 0.05 (specifically, 0.6155, 0.8086, and 0.8246, respectively). These high P-values indicate no statistically significant difference in the mean performances of the models based on the dataset under consideration. Despite RFSH achieving a higher accuracy in practice, the ANOVA results suggest that the observed performance improvements over RFC and RFJS may not be statistically significant at the 95% confidence level. This could be attributed to the relatively small test set size or the inherent variability in model predictions across different folds.

Table 5: Result of statistical analyses based on ANOVA

Model	F-value	P-value
RFC	0.2523	0.6155
RFSH	0.0587	0.8086
RFJS	0.0491	0.8246

4 Discussion

The results of this study indicate that both optimized models, including RFSH and RFJS, outperform the baseline RFC in phishing website classification. The RFSH model achieved the best performance with an accuracy of 95.20%, followed by RFJS (93.20%) and RFC (91.06%). These gains are attributed to the capability of the metaheuristic optimizers to fine-tune model parameters, improving convergence, reducing overfitting, and enhancing generalization.

The superior performance of RFSH compared to RFJS may stem from the SHO's more effective balance between exploration and exploitation. The SHO mimics the spatially aware swimming patterns of sea-horses, which may help it escape local optima better than the JSOA, whose dynamics are influenced by time-varying ocean currents and may be more susceptible to premature convergence.

Table 6 presents a comparative analysis of the approach against state-of-the-art phishing detection models from recent studies. While the Extra Trees Classifier by Awasthi and Goel [46] reports a marginally higher accuracy (98.59%), other ensembles and gradient-boosted models demonstrate performance comparable to or lower than the proposed RFSH model. For instance, Gradient Boosting (GB) in Pandey et al. [47] yielded 95.7%, only slightly higher than our result, whereas XGBoost and GBDT methods in Yang et al. [48] achieved 88.46% and 89.04%, respectively.

Table 6: Comparison of the current work with state of art phishing prediction articles.

Source Article	Model	Result (%)
Awasthi and Goel [47]	Voting ensemble	95.51
	Extra trees classifier	98.59
Yang et al. [48]	GBDT	89.04
	XGB	88.46
Pandey et al [49]	GB	95.7
	DT	94.2
Current Study	RFSH	95.20

The RFSH model offers competitive performance compared to these works, especially considering its relatively low computational complexity and interpretability. Models like Extra Trees may require more extensive feature engineering and hyperparameter tuning, whereas the proposed RFSH balances efficiency and accuracy. Despite these promising results, there are several limitations to consider. First, the dataset was compiled from existing literature and may not fully capture the variability of real-time or zero-day phishing websites. Second, the static feature set may not be adaptable to evolving phishing techniques. Third, the study does not evaluate model latency or deployment feasibility in real-world scenarios. Future work will incorporate real-time URL streams, adaptive feature extraction, and ensemble optimization strategies to improve robustness and scalability further. Additionally, exploring hybrid deep learning approaches and evaluating model performance on live threat feeds could offer deeper insight into practical deployment in cybersecurity systems.

4.1 Real-world application

The proposed RFSH and RFJS models hold strong potential for real-world cybersecurity applications, particularly for detecting phishing websites in real-time. These models can be integrated into security tools such as browser extensions, secure web gateways, email filtering systems, and SIEM/SOAR platforms. Their main advantage lies in their ability to analyze website features and accurately flag suspicious activity quickly. From a computational standpoint, while training the models—especially with optimization algorithms like SHO and JSO—requires moderate to high computational resources, the inference phase (actual prediction) is lightweight and fast. This makes the models suitable for real-time deployment, even on devices with limited resources. Moreover, their compatibility with various programming languages and deployment frameworks allows for easy integration into current infrastructures. They can be deployed via APIs or containerized environments (e.g., Docker) and scaled across cloud-based or enterprise systems. The models are also maintainable, as they support periodic retraining to adapt to new phishing techniques and can be updated remotely. In summary, RFSH and RFJS offer a practical, scalable, and efficient solution for phishing detection in real-time security systems, with minimal latency, strong interoperability, and manageable computational demands.

4.2 Model adaptability and resilience against evolving phishing techniques

Phishing techniques are constantly evolving, making it crucial for detection models to remain adaptive and resilient. While demonstrating strong performance on the current dataset, the proposed model may face challenges when exposed to newly developed phishing tactics that differ significantly from the patterns it has learned. To address this, the model would benefit from periodic retraining using updated datasets that reflect emerging threats and evolving attack vectors. Incorporating a continuous learning framework or active learning mechanism could enhance adaptability, allowing the model to identify novel patterns in real-time and adjust accordingly. Additionally, integrating threat intelligence feeds and community-reported phishing samples could ensure the model remains current and effective against the latest phishing strategies. Acknowledging and preparing for the dynamic nature of phishing threats significantly increases the proposed detection system's practical utility and long-term relevance.

5 Conclusion

This research classifies 1353 websites using machine learning (RFC model) and optimized schemes (Sea Horse and Jellyfish). It enhances technology for privacy and security, preventing information theft. The database is divided into subsets, and accuracy is assessed using metrics. Ultimately, the findings of this research are summarized as follows, which contribute to significant advancement in technology:

- The presence of an IP address exhibited the most positive impact on the outcome, while the Server Form Handler had the most negative effect. This highlights a substantial discrepancy in the significance of these inputs, emphasizing the critical role of the IP address and the adverse consequences linked to the Server Form Handler.
- The data table analysis indicates that the RFSH consistently outperformed the other two schemes in the training, test, and integration phases. The RFSH model showed a significant superiority of 2.14% over the RFJS model and a notable 4.54% advantage over

the RFC model in overall accuracy appraisal. Notably, the RFSH model accurately predicted a substantial number (1353) of sites with an impressive 95.19% correct anticipation rate, reinforcing its effectiveness in predictive modeling compared to alternative schemes.

- The tabular outcomes were visually represented in figures, enhancing visibility and comprehension. Additional figures delineated the miscategorization rates, revealing 1.41 times more miscategorizations for the RFJS model and 1.86 times more for the RFC model than the RFSH model. Furthermore, the ROC diagram for the superior RFSH model illustrated a Micro Average ROC curve with a significantly larger area, which is evidenced by this.
- Substantiates its role as the primary agent or actor in the superiority of this model.

Authorship contribution statement

Yuan Ma: Writing-Original draft preparation Conceptualization, Supervision, Project administration.
Juan Zhao: Methodology, Software
Yongqiang Zhang: Validation.

Data availability

The scholars will make the raw data supporting this article's conclusions available without undue reservation.

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript displays honest work.

Funding

This work was supported by Scientific and Technological Research Program of Chongqing Municipal Education Commission（Grant No.KJQN202401905）

Ethical approval

All scholars have been personally and actively involved in substantial work leading to the paper and will take public responsibility for its content.

Acknowledgement

This work was supported by Scientific and Technological Research Program of Chongqing Municipal Education Commission（Grant No.KJQN202401905）

References

[1] N. Abdelhamid, "Multi-label rules for phishing classification," *Applied Computing and Informatics*, vol. 11, no. 1, pp. 29–46, 2015.

[2] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," *arXiv preprint arXiv:2009.11116*, 2020.

[3] J. Rashid, T. Mahmood, M. W. Nisar, and T. Nazir, "Phishing detection using machine learning technique," in *2020 first international conference of smart systems and emerging technologies (SMARTTECH)*, IEEE, 2020, pp. 43–46.

[4] L. A. T. Nguyen, B. L. To, and H. K. Nguyen, "An efficient approach for phishing detection using neuro-fuzzy model," *Journal of Automation and Control Engineering*, vol. 3, no. 6, 2015.

[5] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for phishing detection," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 232–247, 2022.

[6] R. A. Malaga, "Search engine optimization—black and white hat approaches," in *Advances in Computers*, vol. 78, Elsevier, 2010, pp. 1–39.

[7] R. Maier and T. Hädrich, "Centralized versus peer-to-peer knowledge management systems," *Knowledge and Process Management*, vol. 13, no. 1, pp. 47–61, 2006.

[8] A. J. Flanagin, "IM online: Instant messaging use among college students," *Communication Research Reports*, vol. 22, no. 3, pp. 175–187, 2005.

[9] E. Kirda and C. Kruegel, "Protecting users against phishing attacks with antiphish," in *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, IEEE, 2005, pp. 517–524.

[10] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Tutorial and critical analysis of phishing websites methods," *Comput Sci Rev*, vol. 17, pp. 1–24, 2015.

[11] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Comput Appl*, vol. 31, pp. 3851–3873, 2019.

[12] R. Srinivasa Rao and A. R. Pais, "Detecting phishing websites using automation of human behavior," in *Proceedings of the 3rd ACM workshop on cyber-physical system security*, 2017, pp. 33–42.

[13] R. S. Rao and A. R. Pais, "An enhanced blacklist method to detect phishing websites," in *Information Systems Security: 13th International Conference, ICISS 2017, Mumbai, India, December 16-20, 2017, Proceedings 13*, Springer, 2017, pp. 323–333.

[14] R. S. Rao and S. T. Ali, "Phishshield: a desktop application to detect phishing webpages through heuristic approach," *Procedia Comput Sci*, vol. 54, pp. 147–156, 2015.

[15] P. Kalaharsha and B. M. Mehtre, "Detecting Phishing Sites--An Overview," *arXiv preprint arXiv:2103.12739*, 2021.

[16] M. N. Alam, D. Sarma, F. F. Lima, I. Saha, and S. Hossain, "Phishing attacks detection using machine learning approach," in *2020 third international conference on smart systems and inventive technology (ICSSIT)*, IEEE, 2020, pp. 1173–1179.

[17] V. Bhavsar, A. Kadlak, and S. Sharma, "Study on phishing attacks," *Int J Comput Appl*, vol. 182, no. 33, pp. 27–29, 2018.

[18] I. Saha, D. Sarma, R. J. Chakma, M. N. Alam, A. Sultana, and S. Hossain, "Phishing attacks detection using deep learning approach," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, 2020, pp. 1180–1185.

[19] H. Kamal, S. Gautam, D. Mehrotra, and M. S. Sharif, "Reinforcement Learning Model for Detecting Phishing Websites," in *Cybersecurity and Artificial Intelligence: Transformational Strategies and Disruptive Innovation*, H. Jahankhani, G. Bowen, M. S. Sharif, and O. Hussien, Eds., Cham: Springer Nature Switzerland, 2024, pp. 309–326. doi: 10.1007/978-3-031-52272-7_13.

[20] M. K. Pandey, M. K. Singh, S. Pal, and B. B. Tiwari, "Prediction of phishing websites using machine learning," *Spatial Information Research*, vol. 31, no. 2, pp. 157–166, 2023.

[21] J. Han, J. Pei, and H. Tong, *Data mining: concepts and techniques*. Morgan kaufmann, 2022.

[22] P. A. Barraclough, G. Fehringer, and J. Woodward, "Intelligent cyber-phishing detection for online," *Comput Secur*, vol. 104, p. 102123, 2021.

[23] S. Kumar, A. Faizan, A. Viinikainen, and T. Hamalainen, "Mlspd-machine learning based spam and phishing detection," in *Computational Data and Social Networks: 7th International Conference, CSoNet 2018, Shanghai, China, December 18–20, 2018, Proceedings 7*, Springer, 2018, pp. 510–522.

[24] M. G. HR, A. MV, G. P. S, and V. S, "Development of anti-phishing browser based on random forest and rule of extraction framework," *Cybersecurity*, vol. 3, no. 1, p. 20, 2020.

[25] Y. Ding, N. Luktarhan, K. Li, and W. Slamu, "A keyword-based combination approach for detecting phishing webpages," *computers & security*, vol. 84, pp. 256–275, 2019.

[26] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, pp. 27–39, 2019.

[27] B. Van Dooremaal, P. Burda, L. Allodi, and N. Zannone, "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, pp. 1–10.

[28] P. A. Barraclough, G. Fehringer, and J. Woodward, "Intelligent cyber-phishing detection for online," *computers & security*, vol. 104, p. 102123, 2021.

[29] R. S. Rao and A. R. Pais, "Two level filtering mechanism to detect phishing sites using lightweight visual similarity approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 9, pp. 3853–3872, 2020.

[30] N. F. Abedin, R. Bawm, T. Sarwar, M. Saifuddin, M. A. Rahman, and S. Hossain, "Phishing attack detection using machine learning classification techniques," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, IEEE, 2020, pp. 1125–1130.

[31] M. N. Alam, D. Sarma, F. F. Lima, I. Saha, and S. Hossain, "Phishing attacks detection using machine learning approach," in *2020 third international conference on smart systems and inventive technology (ICSSIT)*, IEEE, 2020, pp. 1173–1179.

[32] Ö. Kasim, "Automatic detection of phishing pages with event-based request processing, deep-hybrid feature extraction and light gradient boosted machine model," *Telecommunication Systems*, vol. 78, no. 1, pp. 103–115, 2021.

[33] S. Anupam and A. K. Kar, "Phishing website detection using support vector machines and nature-inspired optimization algorithms," *Telecommunication Systems*, vol. 76, no. 1, pp. 17–32, 2021.

[34] S. S. Sirigineedi, J. Soni, and H. Upadhyay, "Learning-based models to detect runtime phishing activities using URLs," in *Proceedings of the 2020 4th international conference on compute and data analysis*, 2020, pp. 102–106.

[35] I. Saha, D. Sarma, R. J. Chakma, M. N. Alam, A. Sultana, and S. Hossain, "Phishing attacks detection using deep learning approach," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, 2020, pp. 1180–1185.

[36] D. M. Linh, H. D. Hung, H. M. Chau, Q. S. Vu, and T.-N. Tran, "Real-time phishing detection using deep learning methods by extensions," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, pp. 3021–3035, 2024.

[37] R. Zaimi, M. Hafidi, and M. Lamia, "A deep learning mechanism to detect phishing URLs using the permutation importance method and SMOTE-Tomek link," *The Journal of Supercomputing*, vol. 80, no. 12, pp. 17159–17191, 2024.

[38] H. Kamal, S. Gautam, D. Mehrotra, and M. S. Sharif, "Reinforcement Learning Model for Detecting Phishing Websites," in *Cybersecurity and Artificial Intelligence: Transformational Strategies and Disruptive Innovation*, H. Jahankhani, G. Bowen, M. S. Sharif, and O. Hussien, Eds., Cham: Springer Nature

Switzerland, 2024, pp. 309–326. doi: 10.1007/978-3-031-52272-7_13.

[39] S. Ariyadasa, S. Fernando, and S. Fernando, "SmartiPhish: a reinforcement learning-based intelligent anti-phishing solution to detect spoofed website attacks," *International Journal of Information Security*, vol. 23, no. 2, pp. 1055–1076, 2024, doi: 10.1007/s10207-023-00778-9.

[40] C. Liu, M. White, and G. Newell, "Measuring the accuracy of species distribution models: a review," in *Proceedings 18th World IMACs/MODSIM Congress. Cairns, Australia*, 2009, p. 4247.

[41] S. K. Ghosh and F. Janan, "Prediction of student's performance using random forest classifier," in *Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management, Singapore*, 2021, pp. 7–11.

[42] L. Breiman, "Random forests," *Mach Learn*, vol. 45, pp. 5–32, 2001.

[43] S. Zhao, T. Zhang, S. Ma, and M. Wang, "Seahorse optimizer: a novel nature-inspired metaheuristic for global optimization problems," *Applied Intelligence*, vol. 53, no. 10, pp. 11833–11860, 2023, doi: 10.1007/s10489-022-03994-3.

[44] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Appl Math Comput*, vol. 389, p. 125535, 2021.

[45] A. Alam *et al.*, "Jellyfish search optimization algorithm for mpp tracking of pv system," *Sustainability*, vol. 13, no. 21, p. 11736, 2021.

[46] M. L. Pasini, J. Yin, Y. W. Li, and M. Eisenbach, "A greedy constructive algorithm for the optimization of neural network architectures," *arXiv preprint arXiv:1909.03306*, 2019.

[47] A. Awasthi and N. Goel, "Phishing website prediction using base and ensemble classifier techniques with cross-validation," *Cybersecurity*, vol. 5, no. 1, p. 22, 2022.

[48] R. Yang, K. Zheng, B. Wu, D. Li, Z. Wang, and X. Wang, "Predicting user susceptibility to phishing based on multi-dimensional features," *Comput Intell Neurosci*, vol. 2022, no. 1, p. 7058972, 2022.

[49] M. K. Pandey, M. K. Singh, S. Pal, and B. B. Tiwari, "Prediction of phishing websites using machine learning," *Spatial Information Research*, vol. 31, no. 2, pp. 157–166, 2023.