

An improved weighted metric genetic algorithm based feature selection approach for classification of phishing URLs

Nandeesh Hallimysore Devaraj¹, Prasanna Bantiganahalli Thimappa¹, Prajna Shivarajappa²,
Parashiva Murthy Basavanapura Muddumadappa¹

¹Department of Computer Science and Engineering, Sri Jayachamarajendra College of Engineering,
JSS Science and Technology University, Mysuru, India

²Master of Computer Applications, Sri Jayachamarajendra College of Engineering, JSS Science and Technology University,
Mysuru, India

Article Info

Article history:

Received Mar 26, 2025

Revised Jul 28, 2026

Accepted Aug 12, 2026

Keywords:

Cyber security
Genetic algorithm
Machine learning
Phishing classifier
Uniform resource locator
Web crawler

ABSTRACT

In recent years, websites have been collecting information from users for various purposes. However, users are often unaware that the collected information is being used only for its intended purpose. Phishing is a common social engineering attack that seeks to deceive users into revealing sensitive information, such as bank credentials or personal details, which can be exploited for malicious activities. Protecting this sensitive data from phishing attacks is critical in data security. Many existing approaches effectively classify phishing attacks but fail to distinguish based on the intent of the webpage. This study addresses this problem by proposing a keyword-specific web crawler focusing on extracting a novel feature similarity index based on web page content. Further, the method employs a genetic algorithm (GA) to select highly discriminating optimized features for the classification of phishing uniform resource locator (URLs). The weighted combination of different metrics is used to enhance the model accuracy. As per the performance analysis, the proposed model achieves 98.69% accuracy and 98.75% fitness value with a low false positive rate (FPR). The model interpretability is measured through Shapley additive explanations (SHAP). Additionally, the proposed method is compared with existing works, and experimental results indicate that the proposed model performance is superior over the existing solutions.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Prajna Shivarajappa
Department of Computer Applications, Sri Jayachamarajendra College of Engineering
JSS Science and Technology University
Mysuru, Karnataka 570006, India
Email: prajnas@jssstuniv.in

1. INTRODUCTION

Phishing is a type of social engineering attack in which attackers impersonate legitimate websites to deceive users and steal sensitive information. Phishing is a commonly used method to breach data unethically using many forms like whaling, vishing, uniform resource locator (URL) phishing, and smishing [1]. Phishers use URLs as a base component for performing phishing attacks since most of the phishing attacks are prompted from URLs. Hence, the proposed work mainly concentrates on classification of URLs as either phishing or non-phishing.

In recent years, phishing has gained significant increase in the number of attacks due to the pandemic reported by the anti-phishing working group (APWG) [2]. The world has witnessed 5 million

attacks in the year 2023, which is the worst year for phishing on records. Also 48.2% of all phishing attacks are exploited with social media platforms. According to the report, phishers mainly target social media, the banking sector, the educational sectors and health care as deceptive platforms.

Life cycle of phishing is annotated in Figure 1. Phishers create a fake URL and disseminate it through email, short message services (SMS) or social media channels. When users click on the fake URLs, it redirects the users to spoofed websites that exactly look alike legitimate websites. Most of the spoofed websites' intention is to capture users' sensitive information. The captured information is used to gain access to legitimate websites, which results in financial loss, information loss and reputational loss. Hence, protecting such sensitive and valuable information is crucial in today's digital world.

The remaining part of the paper is structured as follow. Section 2 focuses on related works and highlighting the contributions. Section 3 discusses about the proposed method. Section 4 briefs about the design methodologies adopted in the proposed system. Results are discussed in section 5. Followed by conclusion and future scope in section 6.

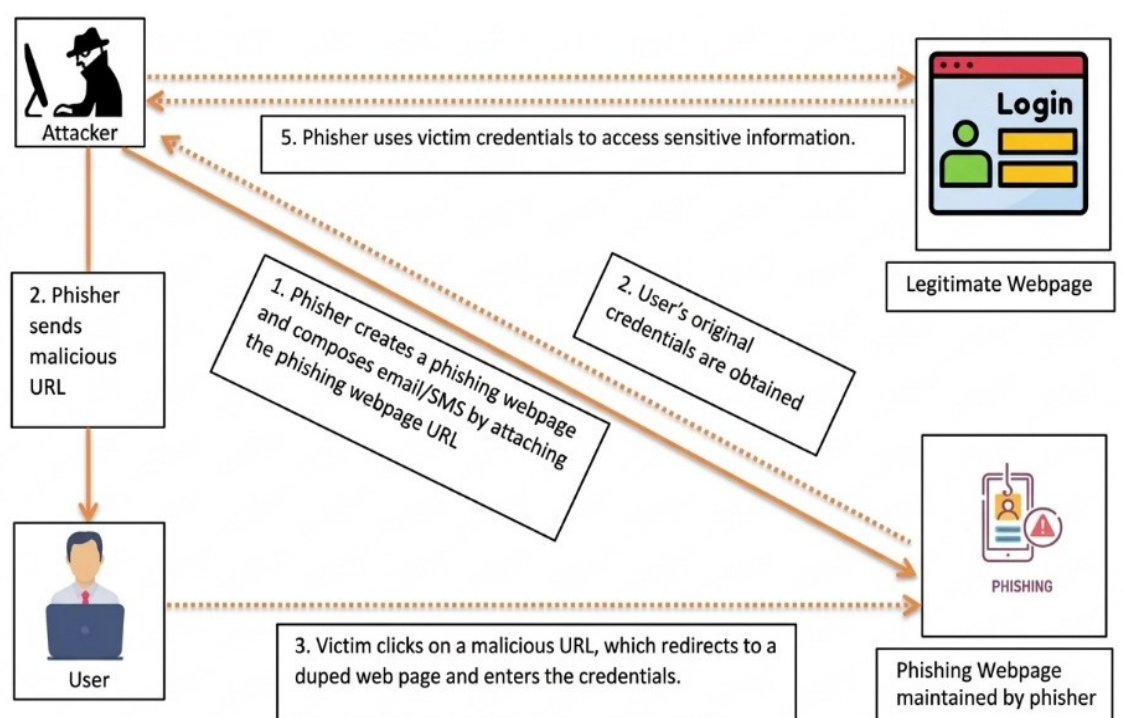


Figure 1. Different stages of phishing life cycle

2. RELATED WORK

URL classification problem can be solved by different approaches. Several authors have used blacklist and whitelist methods. The list-based approach consists of a database where all the malicious URLs are listed in blacklist and a whitelist consisting of legitimate URLs along with other attributes. The list method is efficient but requires frequent updating and not suitable for zero-day attacks. The second approach heuristic method based on heuristic thumb rules. Intrusion detection extension for chrome [3] was designed to identify real-time phishing attacks.

The method combines URL features along with domain, content and page attributes. In addition, the authors introduce proactive and lightweight rule-based incremental construction approach to classify phishing URL. The approach is able to detect zero day and spear phishing attack and achieved an accuracy of 89.12% and 76.2% respectively. Also, true positive rate of 97.13% and 1.5% of false positive values are obtained.

The PhishShield URL [4] was proposed for website content-based desk-top applications. Several heuristic rules are used to detect the phishing attack. Footer links with null values, zero links in the body of the hyper text markup language (HTML), copyright and title content, and website identity are the few heuristic rules used. Also, zero-hour attacks can be easily detectable using the proposed method. The PhishShield produces 96.57% accuracy with less false positive and false negative rate.

Stacking model-based approach was proposed [5] using random forest (RF), k-nearest neighbor (KNN), and neural networks. The work uses UC Irvine machine learning repository-phishing websites (UCIMLR-PWS) dataset consisting of 11,055 samples with 30 features. Also, the proposed method uses principal component analysis (PCA), Relief-F, and the information gain feature reduction method and condenses the number of features from 30 to 5 features with an accuracy of 97.4%. Execution time is considerably high which is not suitable for resource-constrained devices.

Jain and Gupta [6] implemented 5 cross fold validation with 80:20 ratio is divided to 64:16:20 and evaluated with different classifiers. The RF and support vector machine (SVM) classifiers outperform other classifiers. According to performance metrics, RF provides slightly better accuracy of 96.25% and less false positive rate (FPR) value with 4.61% respectively. But the downside is that it is highly dependent on third party services and response time is not considered. The wrapper method with recursive feature elimination (RFE) was exhibited [7] to reduce the dimensionality. The work uses the UCIMLR-PWS dataset with 20 features.

The model is evaluated for different classifiers logistic regression (LR) 89.8%, adaptive boosting (AdaBoost) with 90%, extreme gradient boosting (XGBoost) with 93.2% and gradient boost with 92.8% accuracy. Sabahno and Safana [8] proposed improved spotted hyena optimization (ISHO) algorithm to select proper features. It outperforms the standard spotted hyena optimization (SHO) algorithm in terms of classification. Cost computation is examined between SHO and ISHO. The work uses all 30 features of UCIMLR-PWS dataset. SVM with ISHO yields better performance with accuracy of 98.64%.

Genetic algorithm (GA)-based feature selection approach proposed [9] with deep neural network (DNN) based phishing classifier. GA is used to select the highly optimized set of features with 50 generation each consisting of 20 populations. UCIMLR-PWS dataset used to evaluate DNN model performance. The result demonstrated that a GA based DNN yields accuracy of 88.77%. Additionally, the accuracy is enhanced to 91.13% with GA-based feature weighting with 6 features.

The model is compared with other classifiers such as back propagation neural network (BPNN), SVM, naïve Bayes (NB), C4.5, and KNN, but the result indicates that GA based DNN model performance is superior compared to other classifiers. Bu and Kim [10] proposed a method incorporating the deep learning-based URL classifier with GA to search the optimal feature set that minimizing the false negatives. The author's implemented convolution recurrent neural network (CRNN) model with recall-based GA.

The proposed method achieves 96.85% accuracy and 95.10% of recall with 10-fold cross validation. An enhanced feature selection approach is proposed [11] using GA to detect phishing URL's. Authors proposed a novel feature selection method, based on GAs and built conventional machine learning models to classify URL-based phishing detection system.

Additionally other feature selection methods RFE, sequential feature selection (SFS), recursive features with low variance (RFLV) are also implemented. Model is tested with different feature selection algorithm. As the result indicates, RFLV achieved 91.78 with RF, RFE based RF obtained 91.75 with 45 features, SFS achieves 91.76 accuracy. Proposed GA feature selection obtained 92.93% accuracy. Many existing studies rely on historical data to solve the problem.

In contrast, the proposed approach involves extracting phishing URLs from repositories such as PhishTank and OpenPhish, addressing zero-day attack problem. The proposed work emphasizes on collecting real-time data instead of outdated data samples. The main contributions of the proposed method are:

- i) Designing a keyword specific web crawler to extract the novel similarity index feature from webpage.
- ii) Develop an algorithm which effectively identifies the optimized set of features for classification task.
- iii) Build an efficient machine learning model, which can classify URL as phishing or legitimate.

3. PROPOSED METHOD

The proposed model consists of three phases, as illustrated in Figure 2. The first phase uses the existing UCIMLR-PWS dataset [12] consisting of 11,055 data samples with 30 features related to URL. The proposed work focuses on implementing a keyword-specific web crawler to extract a novel feature similarity score index (SS_Index) from the webpage content of the crawled URLs. The dataset is expanded by incorporating the extracted feature into the existing dataset, along with additional data samples. As a result, the dataset expanded to 18,363 samples, encompassing 31 features in total.

The second phase consisted of data preprocessing and feature selection process. The proposed system uses GA to select the most discriminating optimized set of features from overall feature set for the classification task. The third phase involved with training and testing the different conventional machine learning classifier such as including RF, decision tree (DT), SVM, and KNN for classification process with optimized set of features and model performance is evaluated with different performance metrics.

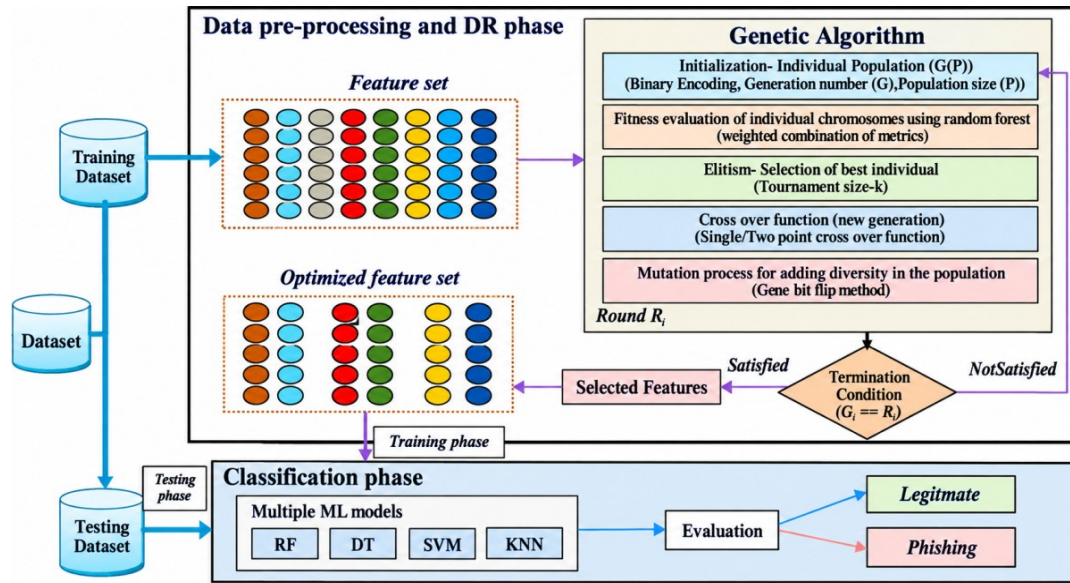


Figure 2. Proposed system for classification of phishing URL's using optimized set of features

4. DESIGN AND METHOD

4.1. Dataset collection and expansion

The proposed system uses UCIMLR-PWS dataset consisting of 11,055 data samples with 30 features. All existing 30 features of UCIMLR-PWS are listed in section 4.3 along with individual feature chi square score and p-values. Further, the dataset has been expanded by crawling legitimate and phishing URLs from Google search engine and PhishTank websites respectively.

A total of 2,000 legitimate data samples related to common phishing target sectors, including healthcare, finance, education, and social media URLs, were gathered from the Google search engine using a keyword-specific web crawler. Additionally, 5,308 general phishing URL samples are collected from PhishTank between October to December 2024. For the crawled URL's, syntactic rules are applied to extract the same set of features which are present in the existing UCIMLR-PWS dataset [13]. Additionally, one more novel feature is extracted from the crawled legitimate URL's and the process is explained in following section.

4.2. Web crawler and novel feature extraction

Web crawler is used by search engines to crawl the web pages and indexing purpose [14]. Web crawler is implemented to collect URL's which are specific to keywords related to health care, social media, finance, and educational sectors. Since these sectors are the most targeted sectors by the phishers as indicated by APWG report. Keywords are inserted as user query to search engine which fetches the related URL's [15]. Each crawled URLs webpage content is parsed to extract words [16]. The similarity score for each retrieved URL is calculated using (1) by comparing the extracted words against bag of words (BoW), which is a pre constructed corpus related to above target sectors. Samples of BoW for a given sector are listed in Table 1.

$$\text{Similarity Score(url)} = \frac{\text{Number of words matched with BoW}}{\text{Total number of words extracted for the given url}} \quad (1)$$

Table 1. Samples of corpus or BoW for each keyword category

Healthcare	Social media	Finance	Education
Appointment	Account	Account	Admission
Check-up	Comment	Amount	Classroom
Consultation	Create	Banking	Curriculum
Doctor	Friend	Branch	Department
Emergency	Photo	Complaint	Examination
Hospital	Public	Credit	Faculty
Medicine	Reel	Deposit	Library
Patient	Share	Money	Result
Registration	User	Transaction	Student
Scan	Video	Username	University

The aim of calculating similarity score is to ascertain whether the URL is relevant to the user's query or not [17]. Phishing webpages are often identified using heuristic thresholds, with one common rule being that if a webpage contains more than 70% external URLs, it is likely to be classified as a phishing page. Hence, the threshold value is set to 70%. In most cases, users are redirected to irrelevant webpages which are not intended for them, particularly in phishing activities. To assess the relevance of a page based on the user's query, a similarity score is computed. The calculated feature value is updated in the database for further processing. The detailed working of web crawler and feature extraction process is presented in Algorithm 1.

Algorithm 1. Calculating similarity score value and SS_Index using a keyword-specific web crawler

Input: $D = \{d_1, d_2, d_3, \dots, d_i\}$: Set of domain related URLs
 K_d : Set of Keywords (BoW/Corpus) for each domain d
 i : Number of domain related keywords selected for crawl
Output: For each URL u , similarity score value between 0 to 1 and a SS_index $S_u \in [0, 1]$.

Step 1: Crawl Top URLs
For each domain $d \in D$:
| Query search engine using keyword specific terms
| Collect top n URLs $\rightarrow U_d = \{u_1, u_2, \dots, u_n\}$

Step 2: Extract and Normalize Keyword for each URL
For each URL $u \in U_d$:
| Fetch homepage HTML content
| Extract visible text and home page keywords
| Let $K_{U_i} = \{K_{U_1}, K_{U_2}, \dots, K_{U_m}\}$ where $m \in \{1 \text{ ton}\}$:
extracted keywords from each URL

Step 3: Compare with Corpus Keywords
Let K_d be the domain keyword corpus: pre-constructed
| Compute the intersection: $M_u = |K_u \cap K_d|$
| Compute the similarity score: $S_u = \frac{M_u}{|K_d|}$

Step 4: Assign SS_Index
Calculate the threshold
| $SS_Index_u = \begin{cases} 1 & \text{if } S_u > 0.70 \\ 0 & \text{otherwise} \end{cases}$

Step 5: Update SS_Index to existing dataset

The proposed algorithm uses a lemmatization function to retrieve the base or root form of extracted keywords. These keywords are compared against the specific corpus to calculate a similarity score for each URL. Based on the similarity score, the novel similarity index value is further derived for each URL to ascertain its relevancy.

4.3. Chi-square based feature significance analysis

To assess the significance of the similarity score novel feature, the proposed system employed a chi-square-based feature ranking method. The chi-square feature ranking algorithm is a statistical technique used to assess the relevance of individual features in relation to a target variable, particularly in classification tasks [18]. The chi-square value for each feature is calculated using (2).

$$\chi^2 = \sum_{T=0}^n \frac{(A_T - P_T)^2}{P_T} \quad (2)$$

Where A_T represents the actual target value and P_T indicates the predicted target value.

Figure 3 illustrates the individual features and their respective significant importance. The similarity score (SS_Index) ranks second in terms of its chi-square score. The ranking indicates that the proposed SS_Index feature is highly discriminative and plays a significant role in the classification of phishing URLs.

Table 2 illustrates each feature predicted chi-square value. The table list chi score value that represents the highest chi-square score and p-value, showing strong correlation with the target variable. Features with very low chi-square scores and higher p-values implies little to no significant relationship with the target variable [19]. A few features demonstrate strong associations with the target variable based on their high chi-square scores and low p-values.

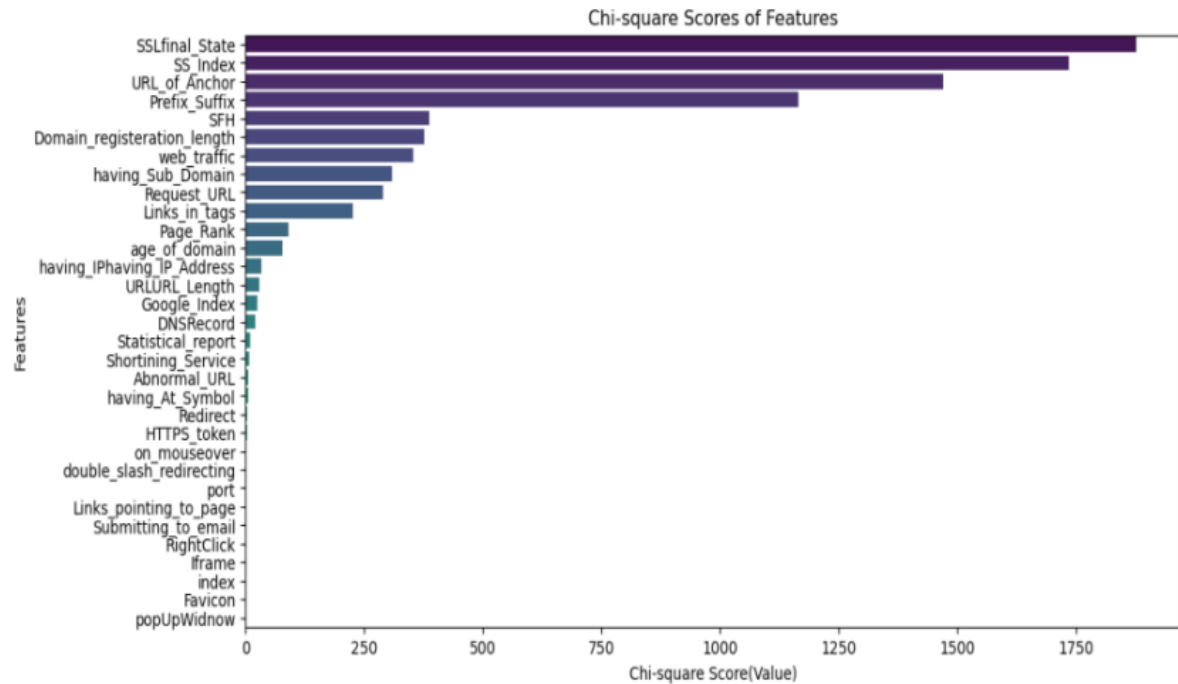


Figure 3. Representation of individual feature chi-square score values

Table 2. Individual features chi square score and p-value

Sl. no	Feature	Chi-square score	P-value
1	having_IP_Address	33.6292	6.6684e-09
2	URL_Length	29.1672	6.6393e-08
3	Shortening_Service	6.6704	9.8029e-03
4	having_At_Symbol	4.6397	3.1240e-02
5	double_slash_redirecting	2.1300	1.4444e-01
6	Prefix_Suffix	1165.4328	1.982e-255
7	having_Sub_Domain	308.9842	3.6351e-69
8	SSL final_State	1876.9047	0.0000e+00
9	Domain_registration_length	376.6977	6.5152e-84
10	Favicon	0.0002	9.8989e-01
11	port	1.9922	1.5812e-01
12	HTTPS_token	2.8526	9.1224e-02
13	Request_URL	288.5678	1.0201e-64
14	URL_of_Anchor	1469.7003	0.0000e+00
15	Links_in_tags	225.3965	6.0164e-51
16	SFH	386.2913	5.3122e-86
17	Submitting_to_email	0.6707	4.1280e-01
18	Abnormal_URL	5.9601	1.4633e-02
19	Redirect	3.9549	4.6735e-02
20	on_mouseover	2.3018	1.2922e-01
21	Right Click	0.0762	7.8250e-01
22	Pop Up Window	0.0000	9.9683e-01
23	Iframe	0.0117	9.1403e-01
24	age_of_domain	76.5968	2.0968e-18
25	DNSRecord	19.7393	8.8756e-06
26	web_traffic	352.3746	1.2883e-78
27	Page_Rank	89.8055	2.6276e-21
28	Google_Index	25.5908	4.2204e-07
29	Links_pointing_to_page	1.4174	2.3383e-01
30	Statistical_report	9.8845	1.6668e-03
31	SS_Index	1734.3716	0.0000e+00

4.4. Genetic algorithm based feature selection approach

The proposed method uses GA for feature selection process [20]. The expanded UCIMLR-PWS with 31 features dataset is split into training and testing samples. The training samples are inputted to GA, which process all 31 features to select the most significant and discriminating features for the classification

task. GA is based on the evolutionary algorithm for feature selection and optimization problem. In the context of phishing URL classification, it helps in reducing the feature set by finding the optimal subset of features that improves the model performance while avoiding unnecessary or redundant features [21]. It selects the optimized set of features as per the below steps.

4.4.1. Initial representation or selection

A subset of features is randomly selected, with 1 representing selected features and 0 for unselected ones [22]. In (3) defines the feature set F for training samples. Each feature values ranging from x_1 to x_n are treating as a gene, and a set of genes forms a chromosome. The population $P^{(0)}$ consists of set of chromosomes $C_i^{(0)}$, generated from the original feature set, as represented in (4).

$$F = \{x_1, x_2, \dots, x_n\} | x_i \in \{0, 1\}; \text{Binary values of feature } f_i \quad (3)$$

$$P^{(0)} = \{C_i^{(0)} = [x_1, x_2, \dots, x_n] | x_j \in \{0, 1\}, 1 \leq j \leq n, 1 \leq i \leq M\} \quad (4)$$

Each individual population represents a candidate subset of feature set F . Binary vector based embedding method is used to represent the population.

4.4.2. Elitism or selection of parents

In this phase, each population is trained with machine learning classifiers to calculate fitness values, which is used to select individuals [23]. The method uses a RF classifier with a weighted combination of metrics (accuracy and F1-score) to evaluate the fitness of individual chromosomes $F(C_i)$ as defined in (5). Parent P_i is selected based on the highest fitness values using the tournament method represented in (6). These selected individuals are used to generate offspring for the next generation. This process is repeated to select M parents for the reproduction process.

$$F(C_i) = \alpha \cdot \text{Accuracy}(C_i) + \beta \cdot \text{F1_score}(C_i); \alpha + \beta \cong 1 \quad (5)$$

$$P_i = \begin{cases} C_a, & \text{if } F(C_a) > F(C_b) \\ C_b, & \text{otherwise} \end{cases} \quad (6)$$

4.4.3. Crossover function

The crossover function simulates biological computation by combining two parent chromosomes to produce better offspring [24]. The approach uses single crossover point method, allowing offspring to inherit the top parent properties, leading to improved performance [25]. The sample chromosomes are depicted in (7) and (8). The process of generating new offspring populations OS_1 and OS_2 using existing chromosomes is represented in (9) and (10). The proposed method selects random crossover point $k \in \{1, 2, \dots, n-1\}$.

$$C_1 = [x_1, x_2, \dots, x_k, x_{k+1}, \dots, x_n] \quad (7)$$

$$C_2 = [y_1, y_2, \dots, y_k, y_{k+1}, \dots, y_n] \quad (8)$$

Offspring generation process:

$$OS_1 = [x_1, x_2, \dots, x_k, y_{k+1}, \dots, y_n] \quad (9)$$

$$OS_2 = [y_1, y_2, \dots, y_k, x_{k+1}, \dots, x_n] \quad (10)$$

4.4.4. Mutation function

The mutation function introduces diversity by flipping a bit 0 to 1 or 1 to 0, preventing local optima and maintaining population variation. The flipped gene is treated as a mutated gene. The algorithm stops after a set number of generations or upon population convergence to an optimal feature subset [26]. For each gene x_i in chromosome C_i , apply mutation with probability P_m . Where x'_j represents the mutated gene, and the formation of mutated gene is given in (11). A mutated chromosome with a mutated gene is represented in (12)

$$x'_j = \begin{cases} 1 - x_j, & \text{with probability } P_m \\ x_j, & \text{otherwise} \end{cases} \quad (11)$$

$$C'_i = [x_1, x_2, \dots, x'_j, \dots, x_n] \quad (12)$$

4.4.5. Termination

To effectively terminate the generation of the population and the calculation of fitness values when no further improvement is detected, the process is governed by a set of predefined criteria, as outlined in (13). These conditions are continuously monitored throughout the optimization process. If any one of these conditions is satisfied—indicating that further iterations are unlikely to yield better results. The algorithm concludes that an optimal feature subset has been found. At this point, the identified optimal feature subset is recorded for classification process, and the generation of new populations is halted to conserve computational resources and prevent unnecessary processing: i) maximum number of generations reached: $t \geq T_{max}$; and ii) convergence (no improvement in fitness over G generation).

$$|F_{best}^{(t)} - F_{best}^{(t-G)}| < \varepsilon \quad (13)$$

Where $F_{best}^{(t)}$ represents the best fitness value at the given generation t and ε indicates the convergence threshold set to 0.30 value based on trial method. GA parameters are fine-tuned for best model performance. Table 3 lists the GA parameters used in the proposed system. These parameters play a pivotal role in deciding model stability and performance.

Table 3. Parameters used in GA

Sl. No	Parameters	Value
1	Population size	50 individuals
2	Generation	50
3	Selection method	Tournament(k-3)
4	Crossover method	Single-point crossover
5	Fitness value (WCM)	0.5* accuracy + 0.5* F1-score

4.5. Learning models

The proposed work is trained and tested with different supervised learning models. SVM, DT, RF, and KNN were adapted for classification purposes [27]. These classifiers are rigorously trained and tested for the classification process. Based on the optimized set of features obtained through the GA, the conventional classifiers are evaluated. For training, 80% of data is used from the dataset and remaining 20% data is used for testing purposes.

5. RESULT AND DISCUSSION

An ablation study was conducted to compare the original dataset with the expanded dataset, focusing on the accuracy of various learning models. When incorporating the SS_Index alongside the existing 30 features, the RF classifier achieved an improved accuracy of 98.55%. In contrast, the same classifier attained 97.29% accuracy using only the original dataset. Table 4 presents the results of different classifiers without the application of a GA. Given that RF outperformed the other classifiers, the area under the curve (AUC) was also computed, resulting in a high AUC score of 0.97. This indicates that the model is effective in confidently classifying phishing instances, even without the use of an optimized feature set.

Table 4. Performance metrics of classifiers for expanded dataset

Classifiers	Accuracy	Precision	Recall	F1-score
RF	98.55	98.30	99.10	98.70
DT	97.33	96.94	98.28	97.60
SVM	96.79	96.15	98.12	97.13
KNN	96.56	95.77	98.12	96.93

AUC: RF-0.97

5.1. Normal fitness value Vs. weighted combination of metrics fitness value

Furthermore, the proposed system incorporates a GA for optimal feature selection, aiming to reduce classification time by using a minimal yet effective subset of features instead of the full feature spectrum. The performance of classifiers SVM, RF, DT, and KNN is evaluated using two types of fitness values: normal fitness value (N-FV) and a weighted combination of metrics fitness value (WCM-FV). Under the N-FV scheme, which considers only accuracy, RF achieved the highest accuracy of 98.64% using 22 features, outperforming the other classifiers. Precision, recall, and F1-score were also calculated, with the results summarized in Table 5. Figure 4 illustrates the best, average, and worst fitness values across generations. A narrow gap between the best (green line) and average (blue line) fitness values from

generation 5 to 50 is noticeable. This limited variation is attributed to misclassifications, as the N-FV metric relies solely on accuracy, neglecting other important metrics such as F1-score or precision. This highlights that excluding these additional metrics for fitness calculation can adversely impact model performance. Results show that RF achieves higher accuracy with GA-based feature selection, reducing features from 31 to 22.

Table 5. Performance metrics of conventional machine learning classifiers with N-FV

Classifiers	Accuracy	Precision	Recall	F1-score
RF	98.64	98.54	99.02	98.78
DT	97.60	97.48	98.20	97.84
SVM	97.01	96.39	98.28	97.33
KNN	96.34	95.75	97.71	96.72

Fitness value 98.64% with 22 features

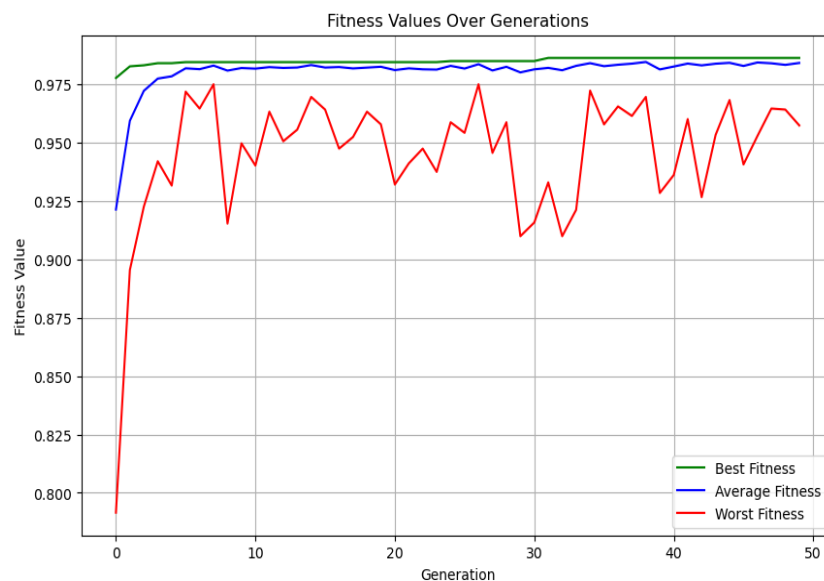


Figure 4. Evolution of best, average, and worst fitness value over number of generations

The fitness value for each generation is computed using the RF classifier. The rationale for selecting RF as the basis for fitness evaluation is illustrated in Figure 5. To ensure a fair comparison, all four conventional classifiers such as RF, SVM, DT and K-NN were evaluated across 10 sample generations. In each of these samples, the RF consistently achieved the highest fitness value when compared to the other classifiers. Furthermore, the best and average fitness values obtained by RF were compared against the best fitness values of the remaining classifiers. The results reveal that the fitness values produced by RF, particularly its best and average scores, significantly outperform those of the other models.

In the second scenario, a weighted combination metric-based fitness value (WCM-FV) is used to evaluate classifiers. To calculate the fitness value, proportion of accuracy and F1-score metrics are used. Instead of using only accuracy, the consideration of F1-score is clearly contributes to enhancing the result of both fitness value and classification accuracy. Results show improved accuracy, with RF classification accuracy increasing from 98.64% to 98.69% and an enhanced fitness value of 98.75%. The WCM-FV demonstrated enhanced classification accuracy when compared against N-FV. Table 6 provides performance metric values of all four classifiers.

Figure 6 depicts the WCM-FV over 50 generations with best, average, and worst fitness values. As the figure depicts, both best and average fitness values are converging after 30 generations. Also, the gap between best and average fitness value is minimized after the 12th generation. For the rest of 50 generation, both best and average fitness values are almost generating similar results. The combination of different metrics for calculation of fitness value is importantly contributing in enhanced fitness value intern selecting best features for the classification process. The enhanced fitness value and classification accuracy clearly reflects the combination of accuracy and F1-score metrics during fitness value calculation. The combination minimizes the FPRs.

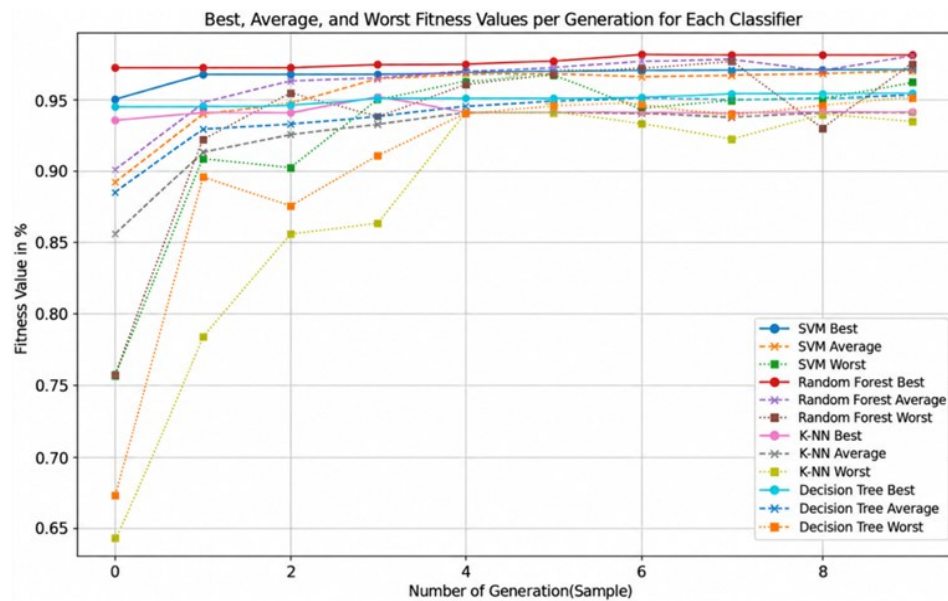


Figure 5. Classifiers best, average, and worst fitness values over number of sample generations

Table 6. Performance of conventional classifiers with WCM-FV

Classifiers	Accuracy	Precision	Recall	F1-score
RF	98.69	98.54	99.10	98.82
DT	96.97	97.07	97.47	97.27
SVM	96.88	96.38	98.04	97.20
K-NN	96.16	95.09	98.12	96.58

Fitness value 98.75% with 22 features

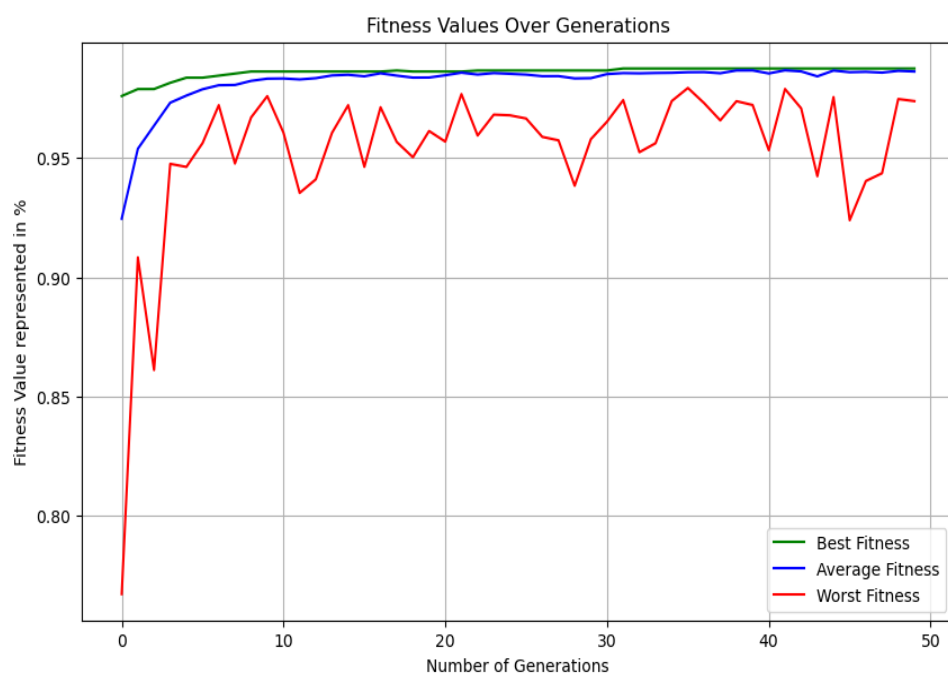


Figure 6. Evaluation of classifiers fitness value based on WCM-FV over generations

To validate robustness, 10-fold stratified cross-validation was conducted, yielding a mean accuracy of 98.68% ($\pm 0.01\%$), indicating stable and consistent performance. A paired two-tailed *t*-test

between training and testing accuracies showed no statistically significant difference ($p = 0.6421$), confirming effective generalization without overfitting. Based on evolutionary principles, the GA selects top 22 most discriminative features out of 31 features and presents them in Table 7. The novel feature SS-Index is selected by GA as a highly discriminative feature and plays a significant role in the classification of phishing URLs.

Table 7. Top 22 features selected by GA for classification process

Sl. No	Feature
1	Having_IP_address
2	Shortening_service
3	Having_at_symbol
4	Prefix_suffix
5	Having_sub_domain
6	SSLfinal_state
7	Domain_registration_length
8	Links_pointing_to_page
9	HTTPS_token
10	Request_URL
11	URL_of_anchor
12	Links_in_tags
13	SFH
14	On_mouseover
15	Iframe
16	Age_of_domain
17	DNSRecord
18	Statistical_report
19	Web_traffic
20	Page_rank
21	Google_Index
22	SS_Index

5.2. Comparative analysis

Figure 7 compares the performance of N-FV and WCM-FV, showing that WCM-FV outperforms N-FV due to the combination of accuracy and F1-score metrics. The WCM-FV method achieved superior performance over the N-FV and attained better classification accuracy. Table 8 illustrates the comparison with existing methods. Ali and Ahmed [9] attained 91.13% accuracy using a GA with DNN, while Sabahno and Safana [8] used an SVM classifier with standard optimization method such as SHO and ISHO methods and achieved an enhanced accuracy of 98.64% over 97.82% accuracy attained by the SHO method. Zamir *et al.* [5] achieved an accuracy of 97.4% using combined stacking models such as RF and neural network along with RFE method.

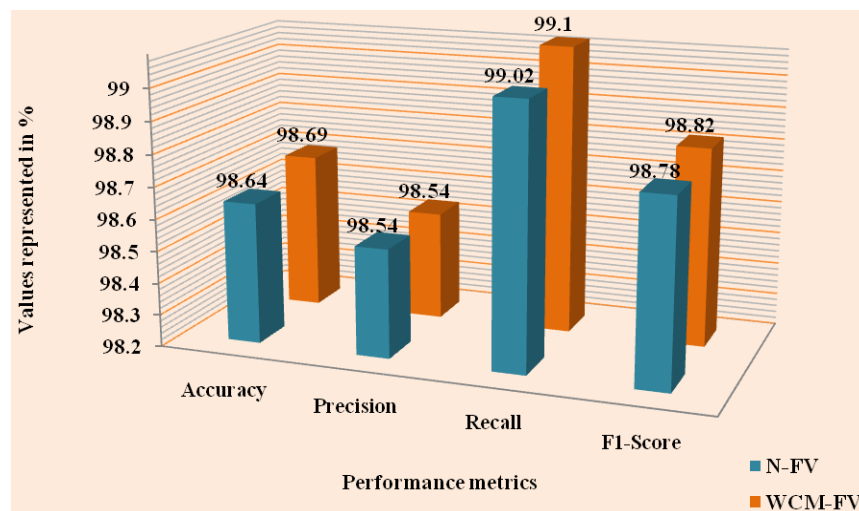


Figure 7. Comparison between N-FV and WCM-FV with different performance metrics

Table 8. Comparison of proposed work results with existing work

Existing works	Accuracy (%)	Precision (%)	Recall (%)	Dataset and features	Method
Zamir <i>et al.</i> [5]	97.4	96	98.1	UCIMLR-PWS - 05 features	RFE with stacking models (RF + neural network + bagging)
Sabahno and Safara [8]	97.82	-	-	UCIMLR-PWS - 30 features	SHO + SVM
Ali and Ahmed [9]	91.13	99.18	97.00	UCIMLR-PWS - 06 features	ISHO + SVM
Proposed model	98.69	98.54	99.10	UCIMLR-PWS - 22 features	GA + DNN GA + RF with SS Index feature

The proposed model clearly outperforms existing approaches in terms of accuracy (98.69%) and recall (99.10%), while maintaining a strong precision score (98.54%). Compared to the best-performing existing models, such as ISHO + SVM accuracy of 98.64% and the RFE-based stacking model recall value of 98.1%, the proposed framework shows improved balance across all evaluation metrics. Moreover, while other existing methods utilize either a very limited set of 5 to 10 features or a relatively complete feature space of 30 features, the proposed approach achieves superior results with a moderate feature set of 22 features. This not only enhances classification performance but also ensures computational efficiency.

5.3. Shapley additive explanations analysis and feature impact interpretation

To enhance model interpretability and explainability, Shapley additive explanations (SHAP) values were employed to quantify the contribution of each feature to the phishing URL classification model's output. The SHAP summary plot presented in Figure 8. The figure illustrates the distribution of SHAP values across all predictions, with features ranked by their mean absolute SHAP value, indicating overall importance.

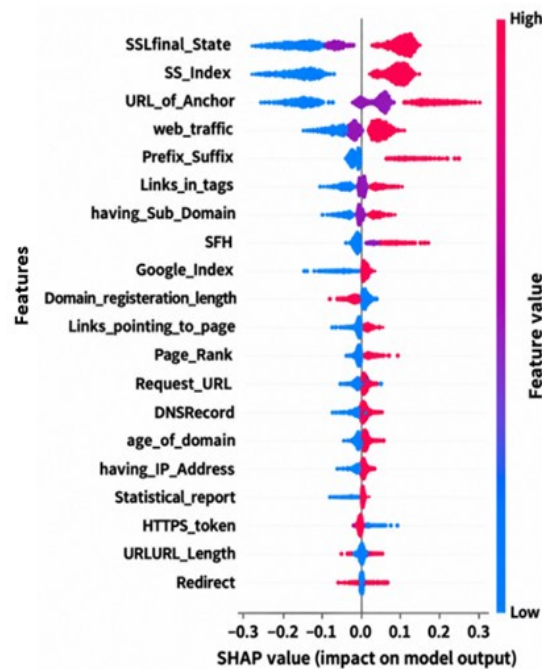


Figure 8. Feature contribution interpretability using SHAP summary value plot

Beyond interpretability, SHAP-based explanations offer actionable insights for practical phishing protection system design. Highly influential features such as SS_Index, SSLfinal_State, and URL_of_Anchor can support confidence-aware alerting, URL prioritization, and adaptive warnings. For example, low SS_Index values combined with deceptive anchor links may trigger high-risk alerts, while ambiguous cases can be escalated for analyst review.

- i) Top influential features: the most impactful features identified include SSLfinal_State, SS_Index, URL_of_Anchor, and web_traffic. These features exhibit a high dispersion of SHAP values, indicating strong and variable influence on model predictions across different samples.

- ii) Directional influence: in the SHAP plot, red (high) and blue (low) feature values reveal their impact on predictions. High values of SSLfinal_State and SS_Index positively influence legitimate classification, while low values lean toward phishing—aligning with domain knowledge. Similarly, high URL_of_Anchor and web_traffic values support legitimacy, whereas low values indicate phishing due to deceptive links and low site visits.
- iii) Moderately important features: features such as having_Sub_Domain, Prefix_Suffix, and Links_in_tags also exhibit noticeable contributions. For example, excessive subdomains and the use of hyphens in domain names (captured by Prefix_Suffix) are commonly associated with phishing behavior and thus negatively impact the classification.
- iv) Less influential features: features like redirect, URL_Length, and HTTPS_token exhibit relatively lower SHAP value magnitudes, indicating limited but potentially case-specific importance.

Overall, the SHAP analysis confirms that the model's predictions align well with established phishing heuristics, providing assurance in the decision-making process and enhancing model transparency. These insights can also guide feature selection and further refinement of phishing detection systems.

6. CONCLUSION

The proposed approach combines a keyword-specific web crawler with a machine learning-based URL classifier. The crawler retrieves URLs based on user queries, and a novel feature similarity score (SS_Index) is computed by extracting keywords from the webpage content. A chi-square-based method ranks the relevance of extracted features, and a GA selects 22 optimal features out of 31. To boost classification performance, a weighted combination metric is applied. Among various classifiers tested, RF achieved the highest accuracy of 98.69% with a fitness value of 98.75%, considering both balanced accuracy and F1-score. Comparative analysis of N-FV and WCM-FV against existing methods shows superior results. SHAP analysis revealed that SS_Index had the second-highest mean absolute SHAP value, highlighting its significant and consistent influence on model predictions. Despite these strengths, the approach is limited by the time overhead of webpage-based similarity computation and its reliance on keyword-specific crawling. Calculating the similarity score involves webpage access and content parsing, which introduces additional computational overhead during the crawling and dataset construction phase. However, this process is performed offline and does not affect real-time URL classification performance. For large-scale deployment, scalability can be improved through parallel crawling, caching mechanisms, and incremental feature updates. Future work will explore keyword-independent crawling and lightweight content abstraction methods to further reduce computational cost.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Nandeesh	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Hallimysore Devaraj														
Prasanna Bantiganahalli	✓	✓			✓	✓		✓		✓		✓		
Thimappa														
Prajna Shivarajappa	✓	✓	✓		✓		✓	✓	✓	✓	✓			
Parashiva Murthy	✓	✓		✓	✓					✓				
Basavanapura														
Muddumadappa														

- C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis
- I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing
- Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [PS], upon reasonable request.

REFERENCES

- [1] M. Vijayalakshmi, S. M. Shalinie, M. H. Yang, and U. R. Meenakshi, "Web phishing detection techniques: a survey on the state-of-the-art, taxonomy and future directions," *IET Networks*, vol. 9, no. 5, pp. 235–246, 2020, doi: 10.1049/iet-net.2020.0078.
- [2] Anti-Phishing Working Group, "Report on phishing activity trends," *apwg.org*. Accessed: Apr. 3, 2025. [Online]. Available: <https://apwg.org/trendsreports/>
- [3] M. S. Kumar, K. G. Srinivasagan, and G. UnniKrishnan, "A lightweight and proactive rule-based incremental construction approach to detect phishing scam," *Information Technology and Management*, vol. 23, no. 4, pp. 271–298, 2022, doi: 10.1007/s10799-021-00351-7.
- [4] R. S. Rao and S. T. Ali, "PhishShield: a desktop application to detect phishing webpages through heuristic approach," *Procedia Computer Science*, vol. 54, pp. 147–156, 2015, doi: 10.1016/j.procs.2015.06.017.
- [5] A. Zamir *et al.*, "Phishing web site detection using diverse machine learning algorithms," *Electronic Library*, vol. 38, no. 1, pp. 65–80, 2020, doi: 10.1108/EL-05-2019-0118.
- [6] A. K. Jain and B. B. Gupta, "Phish-safe: URL features-based phishing detection system using machine learning," in *Cyber Security*, 2018, vol. 729, pp. 467–474, doi: 10.1007/978-981-10-8536-9_44.
- [7] N. S. Goud and A. Mathur, "Feature engineering framework to detect phishing websites using URL analysis," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 295–303, 2021, doi: 10.14569/IJACSA.2021.0120733.
- [8] M. Sabahno and F. Safara, "ISHO: improved spotted hyena optimization algorithm for phishing website detection," *Multimedia Tools and Applications*, vol. 81, no. 24, pp. 34677–34696, 2022, doi: 10.1007/s11042-021-10678-6.
- [9] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," *IET Information Security*, vol. 13, no. 6, pp. 659–669, 2019, doi: 10.1049/iet-ifs.2019.0006.
- [10] S. J. Bu and H. J. Kim, "Optimized URL feature selection based on genetic-algorithm-embedded deep learning for phishing website detection," *Electronics*, vol. 11, no. 7, 2022, doi: 10.3390/electronics11071090.
- [11] E. Kocyigit, M. Korkmaz, O. K. Sahingoz, and B. Diri, "Enhanced feature selection using genetic algorithm for machine-learning-based phishing URL detection," *Applied Sciences*, vol. 14, no. 14, 2024, doi: 10.3390/app14146081.
- [12] R. Mohammad and L. McCluskey, "Phishing websites," *UCI Machine Learning Repository*. 2015, doi: 10.24432/C51W2X.
- [13] P. Bhowmik and P. C. Bhowmik, "A machine learning approach for phishing websites prediction with novel feature selection framework," in *Proceedings of International Conference on Fourth Industrial Revolution and Beyond 2021*, 2022, vol. 437, pp. 357–370, doi: 10.1007/978-981-19-2445-3_24.
- [14] N. Kumar and D. Aggarwal, "Learning-based focused WEB crawler," *IETE Journal of Research*, vol. 69, no. 4, pp. 2037–2045, 2023, doi: 10.1080/03772063.2021.1885312.
- [15] M. Khonji, Y. Iraqi, and A. Jones, "Enhancing phishing e-mail classifiers: a lexical URL analysis approach," *International Journal for Information Security Research*, vol. 3, no. 1, pp. 236–245, 2013, doi: 10.20533/ijisr.2042.4639.2013.0029.
- [16] Y. Fang, Y. Yao, X. Lin, J. Wang, and H. Zhai, "A feature selection based on genetic algorithm for intrusion detection of industrial control systems," *Computers and Security*, vol. 139, 2024, doi: 10.1016/j.cose.2023.103675.
- [17] A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards lightweight URL-based phishing detection," *Future Internet*, vol. 13, no. 6, 2021, doi: 10.3390/fi13060154.
- [18] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi-class SVM," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017, doi: 10.1016/j.jksuci.2015.12.004.
- [19] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings*, 2015, pp. 1200–1205, doi: 10.1109/MIPRO.2015.7160458.
- [20] A. N. S. Charan, Y. H. Chen, and J. L. Chen, "Phishing websites detection using machine learning with URL analysis," in *2022 IEEE World Conference on Applied Intelligence and Computing, AIC 2022*, 2022, pp. 808–812, doi: 10.1109/AIC55036.2022.9848895.
- [21] N. H. Devaraj, P. B. Thimappa, and P. Shivarajappa, "Phish-view: a feature-rich meta ensemble learning for accurate and uncertainty-aware phishing detection," *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 7, pp. 190–209, 2025, doi: 10.22266/ijies2025.0831.14.
- [22] M. Abutaha, M. Ababneh, K. Mahmoud, and S. A. H. Baddar, "URL phishing detection using machine learning techniques based on URLs lexical analysis," in *2021 12th International Conference on Information and Communication Systems, ICICS 2021*, 2021, pp. 147–152, doi: 10.1109/ICICS52457.2021.9464539.
- [23] S. Ghareeb, M. Mahyoub, and J. Mustafina, "Analysis of feature selection and phishing website classification using machine learning," in *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, 2023, pp. 178–183, doi: 10.1109/DeSE58274.2023.10099697.
- [24] X. Liu and J. Fu, "SPWalk: similar property oriented feature learning for phishing detection," *IEEE Access*, vol. 8, pp. 87031–87045, 2020, doi: 10.1109/ACCESS.2020.2992381.
- [25] R. R. Rout, G. Lingam, and D. V. L. N. Somayajulu, "Detection of malicious social bots using learning automata with URL features in Twitter network," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 1004–1018, 2020, doi: 10.1109/TCSS.2020.2992223.
- [26] A. Maci, A. Santorsola, A. Coscia, and A. Iannacone, "Unbalanced web phishing classification through deep reinforcement learning," *Computers*, vol. 12, no. 6, 2023, doi: 10.3390/computers12060118.
- [27] S. S. M. M. Rahman, F. B. Rafiq, T. R. Toma, S. S. Hossain, and K. B. B. Biplob, "Performance assessment of multiple machine learning classifiers for detecting the phishing URLs," in *Data Engineering and Communication Technology*, Singapore: Springer, 2020, pp. 285–296, doi: 10.1007/978-981-15-1097-7_25.

BIOGRAPHIES OF AUTHORS

Dr. Nandeesh Hallimysore Devaraj     presently working as assistant professor in Department of Computer Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India. He received a Ph.D. from JSS Science and Technology University, Mysuru. His general research interest is in the area of information and cyber security, URL phishing detection, web security, mobile security, online social networks, and machine learning. He can be contacted at email: hdnandeesh@jssstuniv.in.



Dr. Prasanna Bantiganahalli Thimappa     received Ph.D. degree from Visvesvaraya Technological University, Karnataka, India in the area of cloud security. He has published more than 60 research articles in international journals and conferences of high repute, including IEEE, Elsevier, and Springer. He is serving as reviewer of Elsevier, IEEE, and many reputed journals. Also, he is a lifetime member of the Computer Society of India (CSI). At present, he is working as an associate professor in the Department of Computer Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India. He can be contacted at email: prasannabt@jssstuniv.in.



Dr. Prajna Shivarajappa     is an assistant professor at JSS Science and Technology University's Department of Computer Applications in Mysuru. She earned Ph.D. from the University of Mysore, researching facial recognition in celebrity cartoon images. She specializes in image processing, pattern recognition, machine learning, and deep learning. She has published several research papers and received a best paper award in many conferences. She can be contacted at email: prajnas@jssstuniv.in.



Dr. Parashiva Murthy Basavanapura Muddumadappa     presently working as an assistant professor in Department of Computer Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India. He received Ph.D. from Visvesvaraya Technological University. His general research interest is in the area of Information retrieval and coding, data analytics, security and machine learning. He can be contacted at email: parashivamurthy@sjce.ac.in.