Signal, Image and Video Processing
https://doi.org/10.1007/s11760-024-03204-2

ORIGINAL PAPER

Detection of phishing URLs with deep learning based on GAN-CNN-LSTM network and swarm intelligence algorithms

Abbas Jabr Saleh Albahadili[1] · Ayhan Akbas[2] · Javad Rahebi[3]

Received: 26 September 2023 / Revised: 25 March 2024 / Accepted: 5 April 2024

Abstract
Phishing attacks are one of the challenges of the Internet and its users. Phishing attacks are an example of social engineering attacks based on deceiving users. In phishing attacks, fake pages that are very similar to legitimate pages are created on the Internet. In phishing attacks, the victim is directed to fake pages, and their valuable information is stolen. Most of the targets of phishing attacks include online payment services, banking, and online sales, so the losses of these attacks are significant. One way to detect phishing attacks is to use machine learning and deep learning methods. The challenge of machine learning and deep learning methods is intelligent feature selection. The lack of feature extraction and intelligent feature selection reduces the accuracy of learning methods in detecting phishing attacks. This paper presents a combined method with deep learning, machine learning, and swarm intelligence algorithms to detect phishing attacks. In the first phase, the dataset is balanced by deep learning based on the GAN. In the second step, the convolutional neural network extracts the primary features from the links and code of web pages. In the third step, the white shark optimizer algorithm selects the essential features. In the last step, the LSTM neural network classifies the samples. The proposed method has been evaluated on ISCX-URL-2016 and Phishtank datasets for feature extraction and selection. The proposed method's accuracy, precision, and sensitivity in the ISCX-URL-2016 dataset are 97.94, 97.82, and 97.76%, respectively. In the Phishtank dataset, the proposed method has accuracy, precision, and sensitivity of 96.78, 95.67, and 95.71%. The proposed method is more accurate than LSTM, CNN, CNN-LSTM, CNN + GA, DNN, VAE-DNN, and AE-DNN methods in detecting phishing.

Keywords Fake pages · Phishing attacks · Generative adversarial network (GAN) · Convolutional neural network (CNN) · Feature selection · Swarm intelligence algorithm

1 Introduction

With the expansion of the Internet and the spread of social media, security challenges have increased [1]. Phishing attacks exemplify these challenges, posing significant threats to Internet users by exploiting social engineering techniques. In phishing attacks, hackers or phishers forge legitimate websites and upload fake versions onto the Internet, closely resembling authentic sites. These fake pages deceive users by replicating features such as fields for entering user information, including usernames and passwords [1, 2]. By stealing such information, phishers can illicitly access legitimate websites [3]. Phishers utilize various communication tools such as email, chat, phone, and social networks to interact with their victims, sending links to fake pages and persuading them to click through deception and social engineering tactics [4].

Phishing attacks have surged during the Covid-19 pandemic, with widespread reliance on online services during quarantine. Reports indicate that phishing attacks target entities like payment gateways, banking institutions, financial entities, and online sales platforms, resulting in substantial financial losses running into billions of dollars [5, 6]. Threats

✉ Ayhan Akbas
a.akbas@surrey.ac.uk

Abbas Jabr Saleh Albahadili
abbas1jabr2saleh@gmail.com

Javad Rahebi
cevatrahebi@topkapi.edu.tr

[1] Department of Computer Engineering, Cankiri Karatekin University, Cankiri, Turkey

[2] Institute for Communication Systems, University of Surrey, Guildford, UK

[3] Department of Software Engineering, Istanbul Topkapi University, Istanbul, Turkey

Published online: 17 June 2024

associated with phishing attacks on social media have seen a 47% increase from the first to the second quarter of 2022, according to the Anti-Phishing Working Group (APWG) [7].

Various methods, including blacklist, heuristic, visual, machine learning, and deep learning approaches, are employed to detect phishing attacks today [8–12]. Blacklist methods store fake site addresses in databases, requiring searches against these databases for detection. However, challenges such as memory consumption, long search times, and the inability to detect zero-day attacks hinder their effectiveness [13] Heuristic methods use mechanisms like address length and JavaScript codes but suffer from high false-positive rates [14]. Visual methods analyze elements such as logos to identify fake pages but lack accuracy [15, 16]. In contrast, machine learning and deep learning methods can detect zero-day attacks and offer learning mechanisms [17, 18]. Various approaches, including artificial neural networks, support vector machines, decision trees, convolutional neural networks, LSTM networks, and GANs, have been proposed for phishing attack detection [19–24]. Feature extraction and selection pose major challenges in machine learning and deep learning methods [25, 26]

Meta-heuristic methods, inspired by biological, physical, and behavioral phenomena, offer efficient feature selection approaches for phishing attack detection. Examples include the genetic algorithm, particle swarm algorithm, whale optimization algorithm, and spotted hyena optimizer [27–30]. The provided Supplementary Table 1 offers an extensive overview of methodologies and approaches used in phishing detection using machine learning algorithms. It details the advantages, disadvantages, and outcomes of each method, along with the purpose, algorithms employed, and corresponding references. These methodologies aim to enhance Internet security by identifying and mitigating phishing risks, safeguarding users from cyber threats. Evaluation techniques include hybrid ensemble feature selection (HEFS), optimal artificial phishing feature selection, and deep learning-based approaches. Additionally, the table explores the utilization of different algorithms such as long short-term memory (LSTM), convolutional neural network (CNN), and light GBM, showcasing their effectiveness in detecting malicious URLs. The ultimate goal is to develop robust machine learning frameworks capable of accurately identifying phishing attempts and bolstering cybersecurity measures.

Another challenge in detecting phishing attacks is the imbalance between phishing samples and typical datasets. Methods such as SMOTE have been proposed to solve the dataset's imbalance challenge. Deep learning methods such as GAN have also been proposed for the challenge of the lack of training samples. One of the main challenges in detecting phishing attacks is the false-positive alarm rate. Intelligent methods should be used to reduce the false-positive alarm rate.

This paper aims to present a method of detecting phishing attacks with a low error rate to reduce the threats caused by phishing attacks. The proposed method in this paper is a multi-step approach. In the proposed method, samples are balanced by GAN deep learning method. In the next step, the convolution neural network features related to the link and code of the web pages are extracted. In the next step, the white shark optimization algorithm [36] is used to select the main feature. In the final stage, the selected features are considered as the input of the LSTM neural network. The role of the LSTM neural network is to classify fake and legal links. The contribution of our authors and innovation in this article includes the following:

- Balancing the dataset of phishing attacks with the GAN deep learning method
- Feature extraction with an improved version of the CNN
- Providing a binary version of the white shark optimization algorithm
- Practical use of white shark optimization algorithm in feature selection and detection of phishing attacks
- Presenting a hybrid approach of CNN, LSTM, and swarm intelligence to detect phishing attacks

This paper has been prepared and written in several parts. The second part examines the background and related works in phishing. In the third part, a proposed method for detecting phishing attacks is formulated and presented. In the fourth part, the proposed method for detecting phishing attacks is implemented and compared with similar methods. In the fifth part, conclusions and future works are presented.

2 Related works

Reports show that phishing attacks are very effective attacks against Internet users. Fishers do not need much knowledge to carry out phishing attacks. In these attacks, fake websites that look very similar to legal websites are loaded on the Internet. The goal of phishers is to steal user information from fake sites. Estimates like the graph in Supplementary Fig. 1 show that the number of phishing attacks has increased between 2020 and 2021 [37]. The increase in phishing attacks causes many users to become victims of these harmful attacks. The main focus and target of phishing attacks are financial and commercial websites on the Internet with many users. Therefore, the number of phishing victims is significant. The diagram of Supplementary Fig. 2 shows the percentage of phishing attacks in each area of the Internet. The focus on phishing attacks on financial institutions, banks, and payment gateways has made the losses of these attacks significant [38]
.

Reports show that most phishing attacks are against online payment gateways and financial services. Considering that the goal of phishing attacks in most cases is to steal money, the losses of these attacks are expected to be significant. According to the diagram in Supplementary Fig. 2, 33% of phishing attacks are against cloud services, and 21 and 19% are against payment gateways and financial organizations, respectively.

In the era of Covid-19, the number of online users on the Internet has increased, leading to increased security challenges and social engineering attacks. The diagram of Supplementary Fig. 3 shows the share of phishing attacks among the challenges related to social engineering attacks [39].

Reports show that phishing attacks have the largest share among social engineering attacks, with a share of 35.3%, and this shows the importance of detecting these attacks. It is essential to understand the process and cycle of phishing attacks to detect phishing attacks. In the diagram of Supplementary Fig. 4, the cycle of phishing attacks is displayed. In phishing attacks, a legitimate website is faked by a fisher, and then fake links are sent to users through communication tools such as email [40].

The detection and prevention of phishing attacks are crucial for safeguarding users' sensitive information. Phishing occurs when users unwittingly engage with malicious links, leading them to fraudulent websites where their credentials are stolen. While traditional approaches like blacklisting and heuristics are common, they may fall short in detecting emerging threats. Machine learning (ML) and deep learning (DL) methods offer a promising solution, particularly in identifying zero-day attacks. In a study conducted in 2023 [41], various DL architectures such as deep neural networks (DNN), recurrent neural networks (RNN), convolutional neural networks (CNN), and recurrent convolutional neural networks (RCNN) were compared for their efficacy in detecting phishing attacks, achieving an accuracy of approximately 84%.

Another study in 2023 [42] explored feature selection methods and ML techniques for phishing detection. The CatBoost classification emerged with the highest accuracy of 97.46%, with principal component analysis (PCA) demonstrating superior feature selection capabilities. Similarly, research in the same year [43] introduced a novel DL method based on LSTM-FCN and BP neural networks for detecting phishing in cryptocurrencies, boasting an accuracy of around 97.86%.

Phishing detection extends beyond web pages to emails and visual cues. In 2023, a study [43] introduced an embedded and hybrid learning method for phishing email detection, achieving an F1 index of approximately 99%. Furthermore, research [44] proposed a technique leveraging common visual and textual identity elements to detect phishing attacks with an accuracy of about 98.6%, notably reducing false positives compared to existing methods.

Moreover, the integration of N-gram-based feature selection with DL architectures such as CNN and LSTM demonstrated promising results, achieving an accuracy of 98.27% [45]. However, challenges persist, particularly in ML's susceptibility to adversarial attacks. To address this, a hybrid DL approach incorporating generative adversarial networks (GANs) was introduced [46], though it showcased reduced accuracy in detecting phishing attempts.

Lastly, innovative approaches such as combining butterfly optimization and harmony search algorithms [47], showcased robust performance, achieving average accuracies of 98.69% and 98.80% on different datasets. These studies collectively underscore the importance of leveraging advanced computational techniques in combating evolving phishing threats and protecting user data integrity.

Examining related works reveals that detecting phishing attacks presents a complex classification problem. In this context, website pages and links serve as inputs, categorized into standard and phishing. Various approaches have been proposed, broadly falling into three categories:

Blacklist method: This approach relies on a list containing addresses of known fake pages to detect phishing attacks. Information on links and fake pages is stored in a database, and web pages or internet links are compared against this list for detection. However, this method faces challenges such as high memory consumption, the need for powerful search algorithms, and the inability to detect zero-day attacks.

Heuristic methods: These methods utilize observations such as address length, the number of dots in the address, and the presence of special characters to identify phishing attempts. Despite their simplicity, heuristic methods suffer from high false-positive (FP) rates, a lack of learning capabilities, and the inability to detect zero-day attacks.

Visual methods: This category involves matching visual elements like website logos with those of legitimate pages. While these methods offer potential, they require image processing and are associated with high complexity and error rates.

Machine learning and deep learning methods: ML and DL techniques offer the ability to detect zero-day attacks with high accuracy. However, they require a balanced dataset to enhance learning model accuracy. Additionally, if the learning process does not encompass fundamental features, these methods may incur a significant error rate. Despite these drawbacks, ML and DL methods remain promising avenues for combating phishing attacks.

3 Methodology

The related works indicate that machine learning and deep learning have the potential to detect new attacks and zero-day phishing attempts. Achieving high accuracy in detecting phishing attacks often requires balancing the dataset by generating artificial samples. Another effective approach is feature engineering, which involves techniques like feature extraction and selection. This research introduces several innovations in phishing attack detection:

- Balancing training and test samples by generating synthetic samples using the GAN deep learning method.
- Introducing a novel approach for converting URL strings into a suitable format for deep learning.
- Extracting features from internet links using a CNN.
- Presenting a binary version of the white shark optimization algorithm to select crucial features in the CNN output.
- Adapting the inputs of the LSTM neural network using features selected by the white shark optimizer (WSO) algorithm to classify samples into two categories: legitimate and phishing.

The remainder of this section outlines the proposed method for detecting phishing attacks, encompassing various components such as dataset balancing with GAN, feature extraction with CNN, feature selection with the WSO algorithm, and sample classification using LSTM.

3.1 The proposed framework

The proposed method outlined in this paper, termed CNN-WSO-LSTM, is specifically designed for the detection of phishing attacks. The framework of this method is visually represented in Fig. 1. According to this framework, the following steps are recommended to effectively detect phishing attacks:

- Set the deep learning parameters for CNN and LSTM, as well as for the WSO algorithm.
- Configure the WSO Algorithm counter, including parameters such as t = 1 and Maxt.
- Collect incoming URLs for analysis.
- Train the GAN deep learning model using training examples.
- Generate artificial URLs with GAN to balance the dataset.
- Train the CNN to extract features from the URLs.
- Encode a feature vector as a member of the WSO algorithm.
- Initialize the population of feature vectors as the initial population of the WSO algorithm.
- Evaluate each feature vector using an MLP neural network based on the average detection error of fake links.
- Determine the fitness of each feature vector based on the average error of detecting phishing attacks and the number of selected features.
- Update feature vectors using the WSO algorithm.
- Repeat the steps of the feature selection algorithm and update the feature vectors with the WSO algorithm iteratively.
- Train the LSTM neural network with the optimal feature vector.
- Evaluate the proposed model's performance in detecting phishing attacks using indicators such as accuracy.

These steps collectively form the methodology proposed for the detection of phishing attacks in this paper, facilitating a comprehensive approach to addressing this critical cybersecurity challenge.

3.2 Balancing dataset

In many instances, the quantity of normal records outweighs the number of phishing samples, resulting in an imbalanced dataset. Addressing this issue is crucial for achieving balance within the dataset. Learning on such an unbalanced dataset can lead to an increased detection error in phishing attacks. To mitigate this challenge, artificial samples are generated using deep learning techniques such as generative adversarial networks (GANs) to balance the dataset. Generating artificial URLs requires a comprehensive understanding of the structure of a URL.

According to studies [48], a URL has 84 different characters such as (a-z, A-Z, 0–9, -. ! ∗ ’();:& = + $,/?#[]). A matrix with 84 rows can be considered, the columns of which are addresses to represent a URL numerically. The structure of this matrix is shown in Eq. (1).

$$\text{Coding(URL)} = \begin{array}{c} \\ \vdots \\ 3 \\ \vdots \\ 8 \\ \vdots \\ 13 \\ \vdots \\ 15 \\ 16 \\ \vdots \\ 20 \\ \vdots \\ 82 \\ 83 \\ 84 \end{array} \begin{array}{c} \overset{https://www.domian.com}{} \\ \vdots \\ \begin{bmatrix} 0\,0\,0 & & 0\,0\,0 \\ 0\,0\,0 & & 0\,0\,0 \\ 0\;\;0\;\;0 & & 1\,0\,0 \\ 1\;\;0\;\;0 & & 0\,0\,0 \\ 0\;\;0\;\;0 & & 0\,0\,0 \\ 0\;\;0\;\;0 & \cdots & 0\,0\,1 \\ 0\;\;0\;\;0 & & 0\,0\,0 \\ 0\;\;0\;\;0 & & 0\,1\,0 \\ 0\;\;0\;\;0 & & 0\,0\,0 \\ 0\;\;0\;\;0 & & 0\,0\,0 \\ 0\;\;1\;\;1 & & 0\,0\,0 \\ \vdots & \ddots & \vdots \\ 0\,0\,0 & & 0\,0\,0 \\ 0\,0\,0 & \cdots & 0\,0\,0 \\ 0\,0\,0 & & 0\,0\,0 \end{bmatrix} \end{array} \tag{1}$$

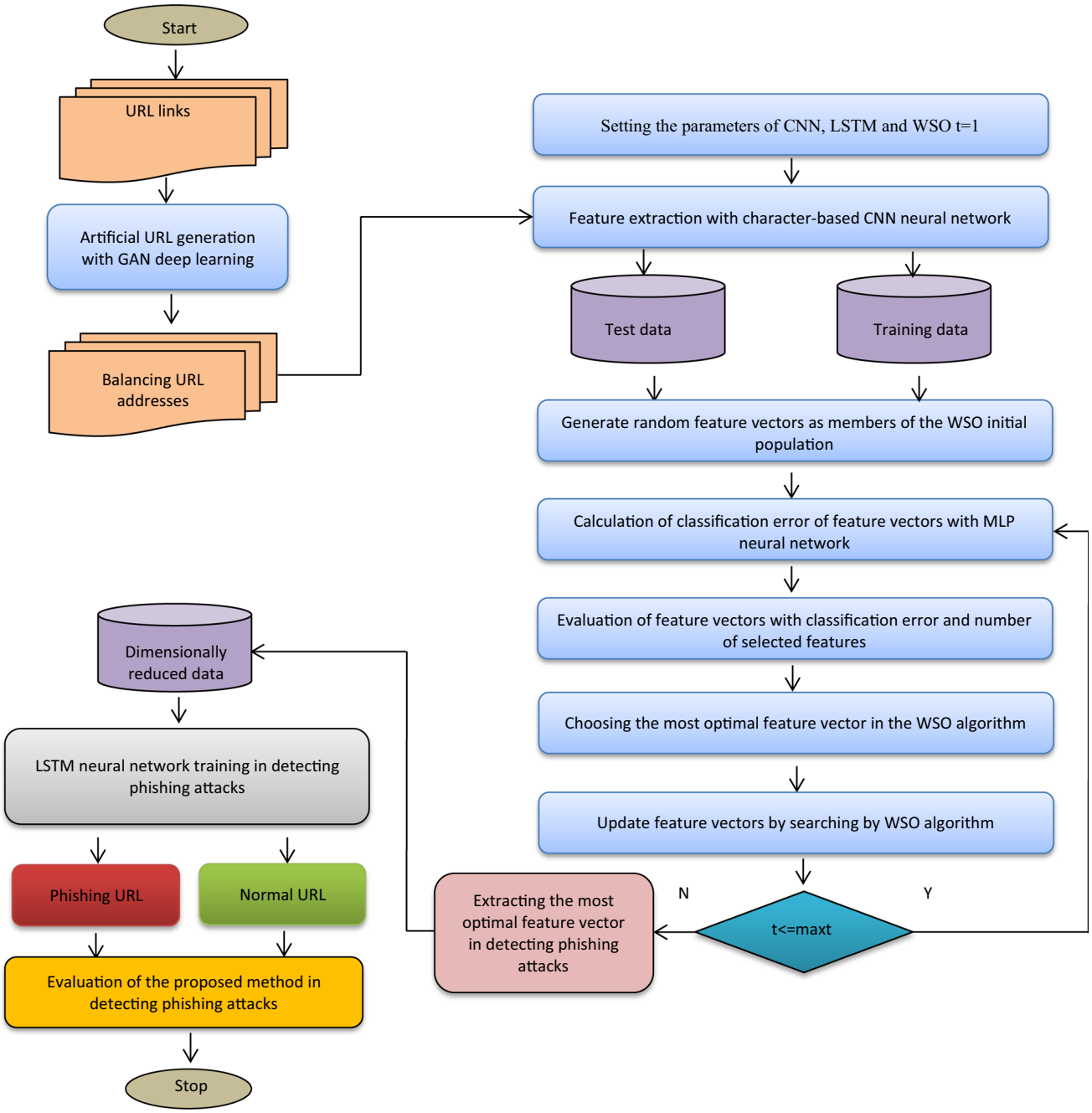

Fig. 1 The framework of the proposed method in detecting phishing attacks

The proposed encoding converts the URL into a numerical matrix comprising zero and one elements. Each "1" in the matrix indicates the mapping of a URL character to a number between 1 and 84. For example, in Eq. (1), the letter "h" is denoted by the number one, corresponding to its position as the eighth letter of the alphabet and located in the first column of row eight. This numerical matrix coding is crucial as it serves as the input for GAN and CNN deep learning methods, ensuring compatibility with their requirements. The generative adversarial network (GAN) plays a pivotal role in generating artificial samples to balance the dataset. Consisting of two models, the generator and discriminator, the GAN competes to examine, capture, and replicate dataset changes. The generator, an artificial neural network, produces fake and artificial URLs with the aim of deceiving the discriminator into considering them legitimate. Conversely, the discriminator's task is to differentiate between fake and artificial data generated by the generator and legitimate URLs (Fig. 2). The process of generating artificial and fake URLs in the GAN involves several steps: creating a noisy and random input

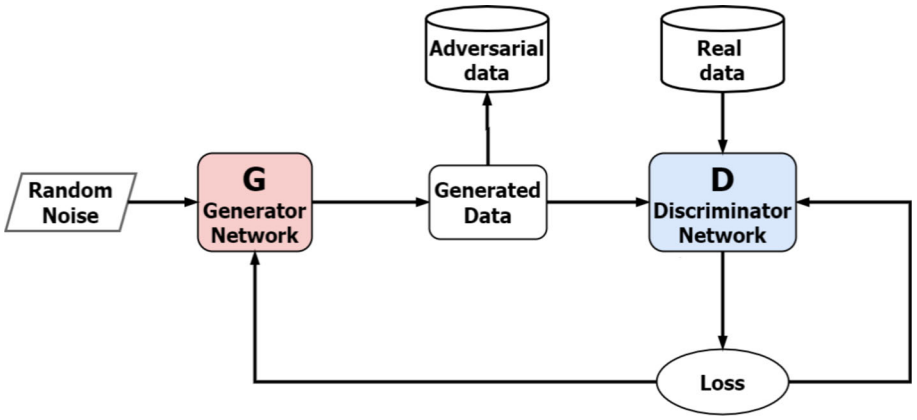

Fig. 2 GAN deep learning structure [49]

vector, transforming the input into a fake sample using the generative network, categorizing the generated data using the discriminant network, and penalizing the generator if the discriminator correctly classifies the fake sample. In the GAN, the generator and discriminator are represented by G and D, respectively, while the real and random inputs are denoted as x and z. The portion of random samples created by G(x) is depicted within the GAN. The discriminator's objective function comprises two components, as outlined in Eqs. (2) and (3).

$$l_{d_1} = \log\sigma(D(x)) \tag{2}$$

$$l_{d_2} = \log(1 - \sigma(D(G(z)))) \tag{3}$$

To maximize the efficiency of the GAN in the discriminator part, the objective function in Eq. (4) needs to be maximized. The numbers inside the log are between zero and one, and for depreciation, it is enough to multiply a negative number in Eq. (4) and minimize the objective function like Eq. (5).

$$L_D = l_{d_1} + l_{d_2} = \log\sigma(D(x)) + \log(1 - \sigma(D(G(z)))) \tag{4}$$

$$L_D = l_{d_1} + l_{d_2} = -(\log\sigma(D(x)) + \log(1 - \sigma(D(G(z))))) \tag{5}$$

The objective function for the generator is placed in Eq. (6), and the objective of minimizing this objective function is:

$$L_G = -\log\sigma(D(G(z))) \tag{6}$$

3.3 Feature extraction

The proposed method employs a convolutional neural network (CNN) at the character level to extract features from the URLs used in the address. The structure of this character-level CNN is detailed in [50], specifically designed for processing text strings and performing classification tasks. The architecture of the convolutional neural network at the character level, as referenced in Fig. 3, illustrates its configuration and operation for this purpose.

In the character-level convolutional networks, six convolution layers are employed, accompanied by three fully connected layers. The weights are initialized using a Gaussian distribution with mean and standard deviation ranging between 0 and 0.2. Initially, the inputs in the CNN network are treated as URLs. Subsequently, as described by Eq. (1), the URL addresses undergo conversion into a numerical matrix, a process similar to that outlined in prior research [51]. This numerical matrix serves as the input for the convolution layers. In the proposed method, the output from the pooling layer is utilized for feature extraction. These extracted features then serve as the input for the LSTM network.

3.4 Feature selection

The output of the feature extraction phase in the proposed method yields a set of extracted features aimed at detecting phishing attacks. However, not all of these features are equally significant; some may be unimportant and could potentially decrease learning accuracy. To address this issue, feature selection is employed to enhance the accuracy of the LSTM model. This phase occurs after the feature extraction phase and is treated as an optimization problem. The feature selection problem is inherently an optimization problem, and it can be effectively tackled using meta-heuristic methods. One such method is the white shark optimization (WSO) algorithm, introduced in 2022 as a form of swarm intelligence. The WSO algorithm [36] offers several advantages:

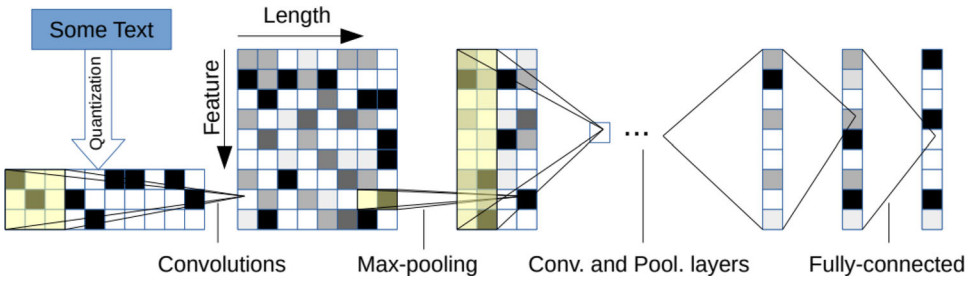

Fig. 3 Character-level convolutional networks structure [50]

- It exhibits higher accuracy in finding optimal solutions compared to popular meta-heuristic algorithms such as GA and PSO.
- It balances national and local searches, facilitating the efficient exploration of the problem space.
- The swarm intelligence behavior embedded within the WSO algorithm enables parallel searching of the problem space, reducing the risk of being trapped in local optima.
- Despite its recent introduction, the capabilities of the WSO algorithm in feature selection have been largely overlooked.

The WSO algorithm is inspired by the hunting behavior of white sharks in nature, particularly in their quest for food in the deep ocean. This algorithm effectively models the exploratory search and exploitation process observed in white sharks. Figure 4 illustrates the process, depicting a white shark searching for food within its surrounding environment.

The hunting behavior of sharks is influenced by distance, leading them to employ various heuristic methods for hunting (Fig. 5). Within a range of less than 50 m, sharks primarily rely on electrical signals to detect the position of their prey. Between distances of 50 and 100 m, they estimate prey positions based on pressure cues, while distances between 100 and 1000 m prompt reliance on olfactory cues. Beyond 1000 m, sharks recognize prey locations based on auditory signals. The white shark optimization (WSO) algorithm is specifically designed to strike a balance between exploratory search behavior and productivity. This algorithm is structured with several steps, detailed and formulated as follows: In the proposed method for feature selection, web page URL feature vectors are encoded as members of the white shark optimization algorithm. Each feature vector is represented as an array of 0s and 1s, denoting the non-selection and selection of features, respectively, in attack detection. The proposed method begins with the random presentation of an initial population of feature vectors, as illustrated in Eq. (7).

$$\mathrm{X} = \begin{bmatrix} X_1^1 & X_1^2 & \dots & X_1^d \\ X_2^1 & X_2^2 & \dots & X_2^d \\ \vdots & \vdots & \vdots & \vdots \\ X_n^1 & X_n^2 & \dots & X_n^d \end{bmatrix} \tag{7}$$

In this equation, the number of elements in the feature vector is denoted as d, and n represents either the number of feature vectors or the initial population of the WSO algorithm. In the proposed method, each feature vector is indexed as x_i, where i denotes the vector number. The component j signifies the dimension of a feature vector, and X represents the initial population of feature vectors. One of the behaviors of the WSO algorithm involves updating the motion of feature vectors with the velocity vector directed toward prey, as depicted in Eq. (8).

$$v_i^{t+1} = \mu\left[v_i^t + p_1\left(X_{\text{gbest}} - x_i^t\right) \times c_1 + p_2\left(X_{\text{pbest}}^i - x_i^t\right) \times c_2\right] \tag{8}$$

In this equation, v_i^t represents the speed of shark number i in iteration t, and v_i^{t+1} denotes the new speed of shark or feature vector i. X_{gbest} represents the most optimal feature vector or prey position, while X_{pbest}^i represents the most optimal position obtained by shark number i. Parameters c1 and c2 are two uniform random numbers between 0 and 1. Additionally, p_1 and p_2 are two coefficients for movement toward the prey and are calculated according to Eqs. (9) and (10):

$$p_1 = p_{\max} + (p_{\max} - p_{\min}) \times e^{-(4t/\text{Maxt})^2} \tag{9}$$

$$p_2 = p_{\min} + (p_{\max} - p_{\min}) \times e^{-(4t/\text{Maxt})^2} \tag{10}$$

The $p_{\min}$ and $p_{\max}$ are set to 0.5 and 1.5, respectively. μ is the contraction coefficient to evaluate the convergence behavior of sharks, and it is calculated according to Eq. (11):

$$\mu = \frac{2}{\left|2 - \tau - \sqrt{\tau^2 - \tau}\right|} \tag{11}$$

Fig. 4 White shark searching for food [30]

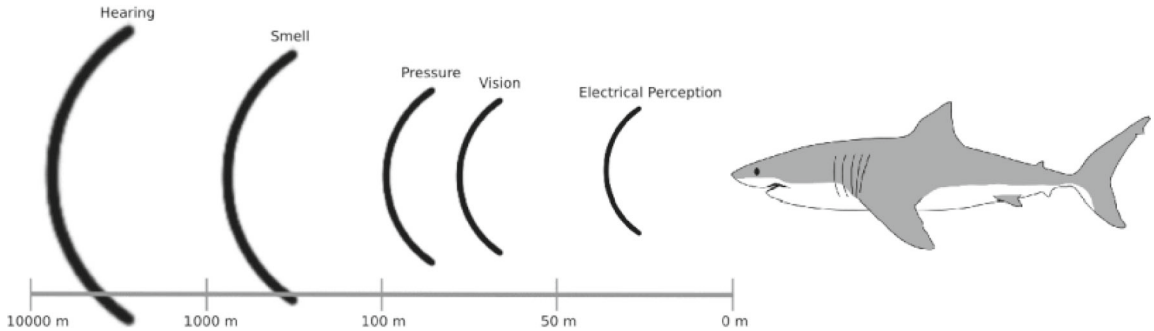

Fig. 5 The heuristic methods of a white shark to hunt prey based on different distances [52]

In this equation, τ stands for the acceleration coefficient of the shark's movement toward the food and is equal to 4.125. The shark moves toward the prey with the calculated speed from the current position and moves according to Eq. (12):

$$x_i^{t+1} = \begin{cases} x_i^t . \neg \oplus x_0 + u.a + lb \text{ rand} < m_v \\ x_i^t + \frac{v_i^t}{f} \text{ rand} \geq m_v \end{cases} \tag{12}$$

Values of a and b are defined as one-dimensional binary vectors. In this problem, the upper and lower boundaries of the search space are denoted by u and l, respectively. ω0 is a rational vector defined in Eq. (15). Equations (13) and (14) are used to calculate Eq. (15).

$$a = \text{sgn}\left(x_i^t - u\right) > 0 \tag{13}$$

$$b = \text{sgn}\left(x_i^t - l\right) > 0 \tag{14}$$

$$x_0 = \oplus(a, b) \tag{15}$$

Values of f represent the frequency of the shark's wavy movement and are formulated according to Eq. (16):

$$f = f_{\min} + \frac{f_{\max} - f_{\min}}{f_{\max} + f_{\min}} \tag{16}$$

m_v is the symbol of the moving force, which increases with the number of rounds that the white shark reaches the prey, as in Eq. (17):

$$m_v = \frac{1}{a_0 + e^{\left(\frac{Maxt}{2} - t\right)/a_1}} \tag{17}$$

In this equation, a_0 and a_1 constants control exploratory search and exploitation. Sharks can move to a position near the prey and, in simpler terms, move to the most optimal shark in the population. For this purpose, Eq. (18) is used.

$$x_i'^{t+1} = x_{\text{gbest}} + r_1 \times \vec{D} \times \text{sgn}(r_2 - 0.5) r_3 < S_s \tag{18}$$

The numbers r_1, r_2, and r_3 are random numbers between zero and one. $\vec{D}$ is the distance between the prey and the shark, formulated in Eq. (19). $x_i'^{t+1}$ is the new position of a shark in line with the optimal solution. S_s is the suggested coefficient to express the strength of the sharks' visual and olfactory senses after chasing other sharks close to the best prey and is formulated in Eq. (20).

$$\vec{D} = \left|\text{rand} \times \left(x_{\text{gbest}} - x_i^t\right)\right| \tag{19}$$

$$S_s = \left|1 - e^{(-a_2 \times t/\text{Maxt})}\right| \tag{20}$$

Equation (21) is used to search for fish school behavior of white sharks.

$$x_i^{t+1} = \frac{x_i'^{t+1} + x_i^t}{2 \times \text{rand}} \tag{21}$$

By running the WSO algorithm, the feature vectors are updated in each iteration. S and V transfer functions are used to make the feature vectors binary, whose rules are formulated in Eqs. (22) and (23).

$$T(x) = \frac{1}{1 + e^{-x}} \tag{22}$$

$$T(x) = \left| \frac{2}{\pi} \arctan\left(\frac{\pi}{2} x\right) \right| \tag{23}$$

The feature vectors are re-normalized using S and V functions between zero and one. According to Eq. (24), if the value of a feature is greater than or equal to 0.5, the feature is selected; otherwise, the feature is not selected.

$$x_i^t = \begin{cases} 1 \; T\left(x_i^t\right) \geq 0.5 \\ 0 \; T\left(x_i^t\right) < 0.5 \end{cases} \tag{24}$$

The evaluation of each feature vector is conducted using an MLP neural network. An optimal feature vector is characterized by producing minimal error in the output of the MLP while containing fewer features. Equation (25) illustrates the feature selection objective function used to assess the feature vectors.

$$\text{Cost}(x_i^t) = \alpha . E + \beta \frac{\left\| x_i^t \right\|}{d} \tag{25}$$

In the given equation, $\|x_i^t\|$ represents the number of selected features in the feature vector x_i^t, while d denotes the dimension of the feature vector. E represents the detection error of phishing attacks by the feature vector x_i^t. In the WSO algorithm, any feature vector that minimizes the value of the objective function holds greater merit.

3.5 Classification with LSTM

The LSTM neural network consists of three gates: the forgetting gate, the input gate, and the output gate. The forgetting gate facilitates the elimination of unnecessary past information, while the input gate is responsible for storing information at time t. Meanwhile, the output gate ensures that not all the information in Ct is transferred to the output h_t . Each gate receives two inputs, namely are x_t and h_{t-1}. These inputs are then subjected to multiplication in two fully connected layers, followed by addition, and finally passing through the sigmoid function. Compared to the RNN network, the LSTM network has four times the parameters and calculation costs. Bidirectional recurrent neural networks, on the other hand, amalgamate two independent neural networks. This configuration enables networks to gather both forward and backward information about the sequence at each time step. Utilizing the Bi-LSTM network facilitates the seamless transition of model and state from future to past and vice versa. In a backward-running LSTM network, future information is retained. Through the Bi-LSTM network, information from both the past and future can be preserved at any given time. Figure 6 illustrates the structure of the Bi-LSTM neural network and its cells.

In Eqs. (26), (27), (28), (29), (30), (31), (32), and (33), the modeling of an LSTM cell is formulated.

$$W = \begin{bmatrix} W_f \\ W_i \\ W_c \\ W_o \end{bmatrix}, \; b = \begin{bmatrix} b_f \\ b_i \\ b_c \\ b_o \end{bmatrix} \tag{26}$$

$$\sigma(\alpha) = \frac{1}{1 + e^{-\alpha}}, \; \tanh(\alpha) = \frac{e^{\alpha} - e^{-\alpha}}{e^{\alpha} + e^{-\alpha}} \tag{27}$$

$$f_t = \sigma\left(W_f \cdot \left[x_t, \, h_{t-1}\right] + b_f\right) \tag{28}$$

$$i_t = \sigma\left(W_i \cdot \left[x_t, \, h_{t-1}\right] + b_i\right) \tag{29}$$

$$\tilde{c}_t = \tanh\left(W_c \cdot \left[x_t, \, h_{t-1}\right] + b_c\right) \tag{30}$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \tag{31}$$

$$o_t = \sigma\left(W_o \cdot \left[x_t, \, h_{t-1}\right] + b_o\right) \tag{32}$$

$$h_t = o_t \times \tanh(c_t) \tag{33}$$

3.6 Pseudocode proposed

The pseudocode of the proposed CNN-WSO-LSTM method for detecting phishing attacks is depicted in Fig. 7. In this method, the initial step involves considering a set of URLs, comprising both fake and legitimate pages, as input for phishing detection. It's worth noting that phishing datasets often suffer from imbalance, with fewer phishing samples compared to legitimate ones. To address this challenge, the GAN deep learning method is employed to generate artificial examples of fake URLs, which are then added to the dataset.

Following dataset preparation, balanced samples are utilized to train the CNN, which subsequently extracts primary features. These extracted features are then passed on to the feature selection phase, where essential features are identified using the WSO binary algorithm. The selected features

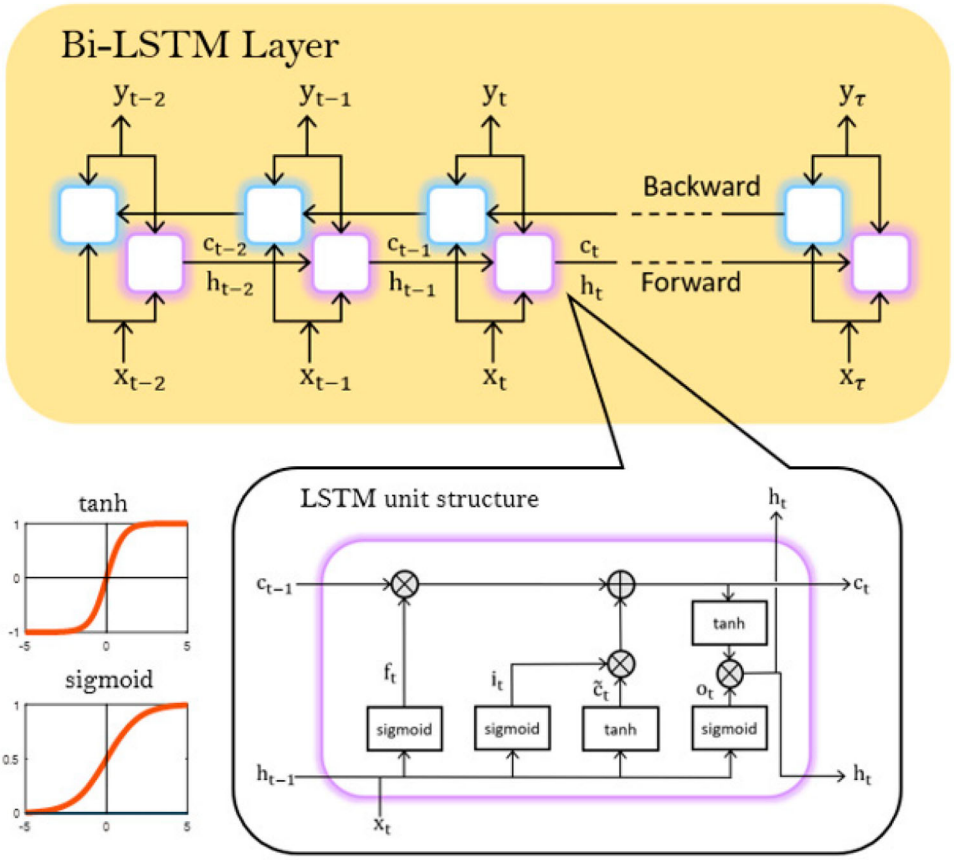

Fig. 6 Bi-LSTM structure and LSTM neural network cell structure [53]

are then forwarded to the LSTM neural network responsible for classifying URLs into phishing and regular categories.

The proposed method for phishing detection is further detailed through the following steps outlined in the pseudocode:

- Normalization of the dataset using the GAN deep learning technique
- Extraction of features using deep learning through the CNN network
- Selection of features employing a binary version of the WSO algorithm
- Reduction of sample dimensions using the optimal feature vector
- Training of the LSTM neural network

4 Analysis

This section implements the proposed method for detecting phishing attacks using MATLAB and Python. The tests conducted in this section involve the implementation of the proposed method and subsequent comparison with similar methods.

4.1 Dataset

In this paper, the set of URLs is gathered from sources including the ISCX-URL-2016 dataset and the Phishtank dataset [50, 56]. To address the issue of dataset imbalance, the number of samples has been balanced using the GAN neural network. Specifically, the dataset now comprises 20,000 legal samples and 20,000 phishing samples. Additionally, the proposed method has generated 10,000 real phishing examples and 10,000 artificial phishing examples. As part of the proposed method, common URL components such as as "http://", "https://" and "www" have been removed.

4.2 Evaluation metrics

To assess the effectiveness of the proposed method, evaluation indices including accuracy, sensitivity, and precision are utilized. These indices are formulated according to Eqs. (34), (35), and (36).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{34}$$

$$\text{Sensitivity} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{35}$$

Training CNN *network on URLs for feature extraction*

Coding feature vectors *as population members of the* ***WSO*** *algorithm*

Generate feature vectors *as random population from* ***WSO*** *algorithm*

Evaluation of feature vectors with the objective function of feature selection and selection of the most optimal feature vector

while (t< Maxt) do

Update the parameters ***v, p1, p2, μ, a, b, w0, f, mv*** *and* ***Ss***

For *i = 1* ***to*** *n* ***do***

$$v_i^{t+1} = \mu[v_i^t + p_1(X_{gbest} - x_i^t) \times c_1 + p_2(X_{pbest}^i - x_i^t) \times c_2]$$

end for

for *i = 1* ***to*** *n* ***do***

if *rand <* ***mv*** *then*

$$x_i^{t+1} = x_i^t . \neg \oplus x_0 + u.a + lb$$

else

$$x_i^{t+1} = x_i^t + \frac{v_i^t}{f}$$

end if

end for

for *i = 1 to n* ***do***

if *rand* $\leq$ *Ss* ***then***

if *i == 1* ***then***

$$x_i^{t+1} = x_{gbest} + r_1 \times \vec{D} \times sgn(r_2 - 0.5)$$

else

$$x'^{t+1}_i = x_{gbest} + r_1 \times \vec{D} \times sgn(r_2 - 0.5)$$

$$x_i^{t+1} = \frac{x'^{t+1}_i + x_i^t}{2 \times rand}$$

end if

end if

end for

Evaluate and update the new feature vectors

t = t + 1

end while

LSTM training *with optimal feature vector and evaluation of the proposed model*

Fig. 7 Pseudocode of the proposed CNN-WSO-LSTM method

Table 1 Evaluation metrics of the proposed method and deep learning on the ISCX-URL-2016 dataset

Method	Accuracy	Precision	Sensitivity
LSTM	95.82	94.63	95.44
CNN	95.82	94.83	93.39
CNN-LSTM	96.14	95.54	94.21
CNN-WSO-LSTM (proposed Method)	97.94	97.82	97.76

Table 2 Evaluation metrics of the proposed method and deep learning on the Phishtank dataset

Method	Accuracy	Precision	Sensitivity
LSTM	92.21	91.13	90.18
CNN	91.64	89.83	90.08
CNN-LSTM	92.31	91.27	91.14
CNN-WSO-LSTM (proposed Method)	96.78	95.67	95.71

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{36}$$

TP, TN, FP, and FN parameters are defined as follows:

- True positive (TP): The URL exampsle is of the phishing type and is classified in the phishing category.
- False positive (FP): The URL sample is of legitimate type and is wrongly classified in the phishing category.
- True negative (TN): The URL sample is of legal type and is classified in the legal category.
- False negative (FN): The URL sample is of phishing type and is wrongly classified in the legal category.

5 Results and discussion

The proposed method has been implemented using MATLAB and Python software, along with libraries such as *Keras* and *Tensorflow*. The training data size is set to 70% of the total data, with 15% allocated for test data and another 15% for validation data. The structure of the convolutional neural network follows the specifications outlined in [46] Dataset normalization is performed within the range [0,1]. The WSO algorithm utilizes a population size of 20 and undergoes 100 repetitions. Parameters c_1 and c_2 are randomly selected from the interval [0,1]. Additionally, the WSO algorithm parameters *pmin* and *pmax* are assigned values of 0.5 and 1.5, respectively, while the coefficient τ is set to 4.125. Tables 1 and 2 display the accuracy, precision, and sensitivity of the proposed method in detecting ISCX-URL-2016 and Phishtank phishing attacks.

The proposed method has been compared with LSTM, CNN, and CNN-LSTM methods for evaluation and comparison purposes. The CNN-LSTM network serves as a model for the proposed method, but without the feature selection phase. To facilitate a comprehensive analysis, the results of the proposed method have been visually compared with those of deep learning methods, as depicted in Figs. 8 and 9. The experiments conducted reveal that in the ISCX-URL-2016 dataset, the LSTM deep learning method achieves an accuracy, precision, and sensitivity of 95.82, 94.63, and 95.44%, respectively. On the other hand, the CNN deep learning method achieves an accuracy, precision, and sensitivity of 95.82, 94.83, and 93.39%, respectively.

The accuracy, precision, and sensitivity of the CNN-LSTM method in the ISCX-URL-2016 dataset for detecting attacks are 96.12, 95.84, and 94.21%, respectively. On the other hand, the proposed method, CNN-WSO-LSTM, achieves an accuracy of 97.94%, precision of 97.82%, and sensitivity of 97.76% in detecting phishing attacks. Compared to the CNN-LSTM method, the proposed method has shown improvements in accuracy, precision, and sensitivity indices by 1.77, 1.01, and 3.55%, respectively, attributed to the intelligent feature selection by the WSO algorithm.

In the Phishtank dataset, LSTM achieves an accuracy, precision, and sensitivity of 92.21, 91.13, and 90.18%, respectively. For CNN, the corresponding metrics are 91.64, 89.83, and 90.08%. CNN-LSTM records an accuracy, precision, and sensitivity of 92.31, 91.27, and 91.14%, respectively. In contrast, the proposed method achieves higher accuracy, precision, and sensitivity of 96.78, 95.67, and 95.71%, respectively, for detecting attacks within the Phishtank dataset. The proposed method integrates deep learning, machine learning, and meta-heuristic algorithms for enhanced performance. To provide a more comprehensive evaluation, the method's efficacy has been compared with modern meta-heuristic algorithms for detecting phishing attacks using MATLAB. Figures 10, 11, and 12 illustrate the comparative analysis of accuracy, precision, and sensitivity indices of the proposed method against several meta-heuristic algorithms. The implementation of swarm intelligence methods such as WOA, HHO, and JSO for feature selection in MATLAB demonstrates superior performance of the proposed method in detecting phishing attacks.

The experiments conducted reveal that the WOA, HHO, and JSO methods achieve accuracies of 96.92, 97.32, and 97.51%, respectively, in detecting phishing attacks. In comparison, the proposed method demonstrates higher accuracy, reaching around 97.94%. Moreover, the precision index of the proposed method for detecting attacks with WOA, HHO,

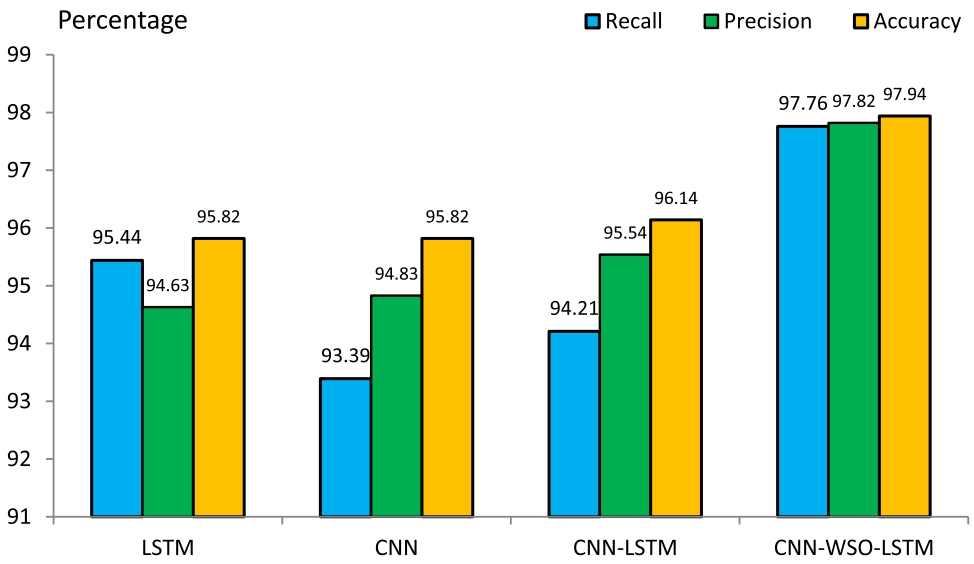

Fig. 8 Comparison of the evaluation metrics of the proposed method with several deep learning methods in the ISCX-URL-2016 dataset

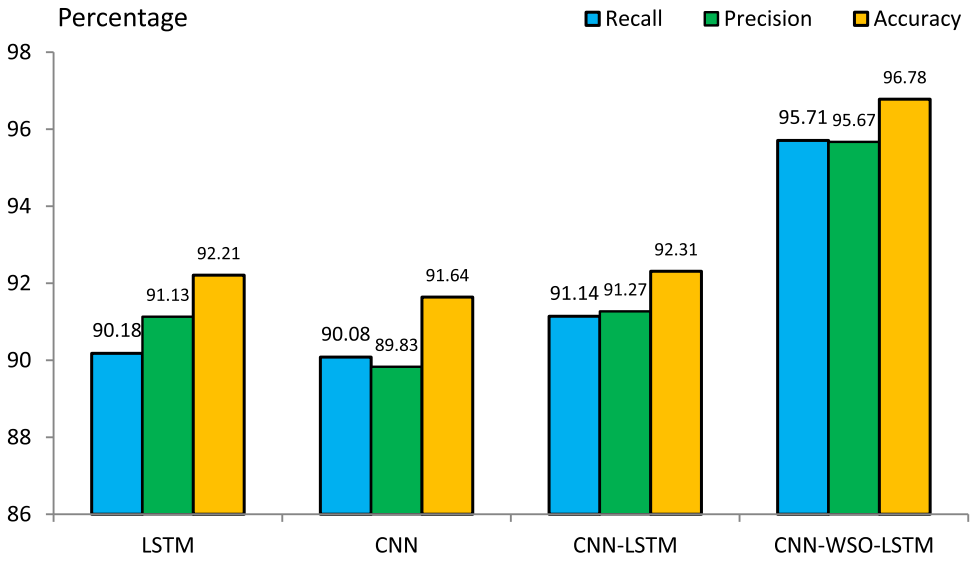

Fig. 9 Comparing the evaluation metrics of the proposed method with several deep learning methods in the Phishtank dataset

and JSO is 96.82, 97.25, and 97.36%, respectively, with an overall accuracy of 97.82%. Similarly, the sensitivity index for WOA, HHO, and JSO is 96.41, 96.64, and 97.26%, respectively, while the proposed method achieves a sensitivity of 97.76%. The evaluation clearly indicates the superiority of the proposed method over WOA, HHO, and JSO algorithms in terms of accuracy, sensitivity, and precision for detecting phishing attacks.

The proposed method exhibits the lowest error rate among feature selection methods for detecting phishing attacks. Conversely, the WOA algorithm demonstrates the poorest performance in these tests. Several factors contribute to the effectiveness of the proposed method in the feature selection phase:

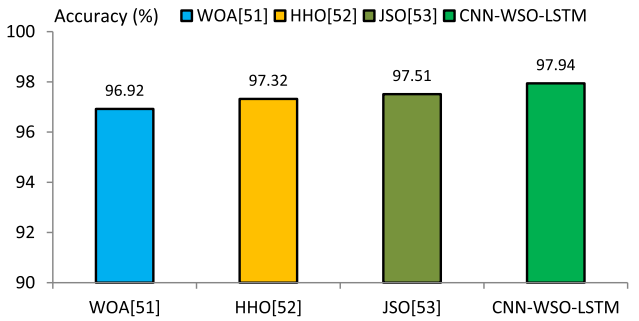

Fig. 10 Comparison of the accuracy index of the proposed method and feature selection methods in the ISCX-URL-2016 dataset

- Unlike the WOA, HHO, and JSO algorithms, the WSO algorithm possesses a velocity vector that enables directional changes according to problem conditions.

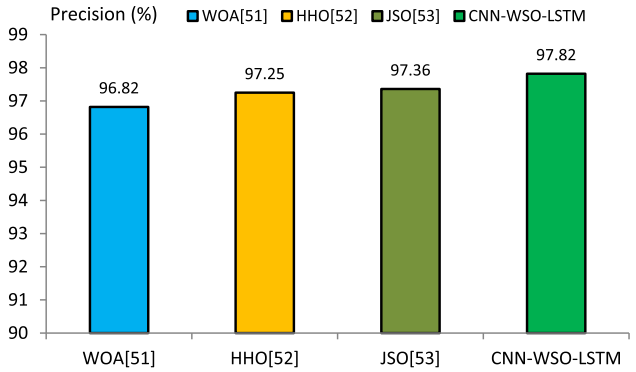

Fig. 11 Comparison of the precision index of the proposed method and feature selection methods in the ISCX-URL-2016 dataset

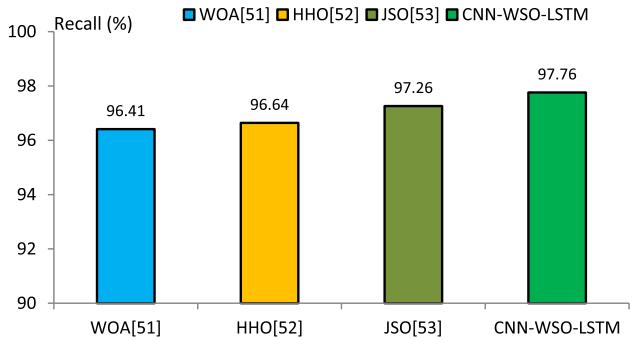

Fig. 12 Comparison of the sensitivity index of the proposed method and feature selection methods in the ISCX-URL-2016 dataset

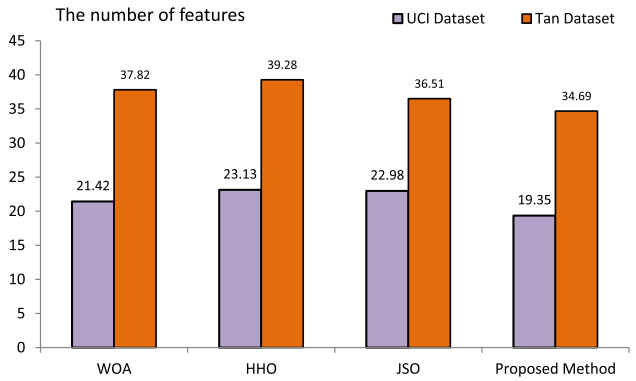

Fig. 13 Comparison of the average number of features selected in the proposed method with feature selection methods

- The WSO algorithm initially employs a global search mechanism before dynamically transitioning to a local search process.
- The diverse search behaviors inherent in the WSO algorithm facilitate thorough exploration of the feature space.

An essential metric for evaluating meta-heuristic algorithms is the number of selected features in detecting phishing attacks. Comparisons have been made between the number of features selected by the proposed method and meta-heuristic methods WOA, HHO, and JSO across different datasets. Unlike the PhishTank dataset, the UCI and Tan datasets require no feature extraction, with 30 and 48 selected features, respectively. Figure 13 illustrates the average number of features selected by the proposed method and WOA, HHO, and JSO algorithms across these datasets.

Table 3 Comparison of the proposed method with deep learning methods in ISCX-URL-2016 and PhishTank

Dataset	ISCX-URL-2016		PhishTank	
Metrics	Acc (%)	Recall (%)	Acc (%)	Recall (%)
Character-CNN [54]	93.63	89.09	88.52	80.34
LSTM [55]	91.75	88.03	85.44	78.65
CNN-LSTM [54]	94.24	90.15	90.70	83.74
URLNet [56]	94.50	93.90	92.26	87.85
Texception Net [57]	97.65	94.62	93.19	90.75
CNN + GA[54]	96.85	95.10	94.83	90.81
Proposed Method	97.94	97.76	96.78	95.71

In the UCI dataset, the proposed method successfully selects 19.35 features out of 30 primary features, while WOA, HHO, and JSO algorithms select 21.42, 23.13, and 22.98 features, respectively. Similarly, in the Tan dataset, the proposed method selects 34.69 features, compared to 37.82, 39.28, and 36.51 features selected by WOA, HHO, and JSO algorithms, respectively. This demonstrates greater dimensionality reduction by the proposed method compared to WOA, HHO, and JSO algorithms in both datasets.

Furthermore, the proposed method's performance in feature extraction, feature selection, and classification is compared with several deep learning methods across the UCI and Tan datasets, as well as with methods presented in 2023 in the ISCX-URL-2016 dataset, as shown in Tables 3 and 4. Notably, the proposed method outperforms character-CNN, LSTM, CNN-LSTM, URLNet, TexceptionNet, and CNN + GA methods in terms of accuracy for detecting phishing attacks in both the ISCX-URL-2016 and PhishTank datasets.

The proposed method in the ISCX-URL-2016 dataset exhibits higher accuracy, precision, and sensitivity compared to AE-DNN, deep AE-DNN, denoising AE-DNN, sparse AE-DNN, convolutional AE-DNN, and contractive AE-DNN methods. Additionally, it outperforms the VAE-DNN method in accuracy and sensitivity indices for detecting phishing attacks. However, it is noteworthy that the VAE-DNN method achieves higher accuracy than the proposed method.

Table 4 Comparison of the proposed method with deep learning methods on the ISCX-URL-2016 dataset

Model	Accuracy (%)	Precision (%)	Recall (%)
AE-DNN [58]	91.45	92.77	90
Deep AE-DNN [58]	93.25	94.39	92.02
Denoising AE-DNN [58]	95.15	96.04	94.22
Sparse AE-DNN [58]	94.85	95.83	93.82
Convolutional AE-DNN [58]	95.91	96.91	94.88
Contractive AE-DNN [58]	96.55	97.02	96.08
VAE–DNN [58]	97.45	**97.89**	97.20
Proposed method [58]	**97.94**	97.82	**97.76**

6 Conclusion and future work

Phishing attacks pose significant challenges for internet users, jeopardizing their sensitive information such as usernames and passwords and costing millions of dollars annually to online services. Detecting such attacks necessitates advanced techniques, particularly machine learning and deep learning methods, which can effectively combat zero-day attacks. However, these methods face numerous challenges, including imbalanced datasets and feature engineering complexities, which can compromise detection accuracy.

This paper introduces a novel approach integrating machine learning, deep learning, and swarm intelligence to address these challenges. The proposed method leverages the GAN deep learning technique to balance fake and original samples and utilizes the character-CNN for feature extraction. It introduces URLs as a numerical matrix for CNN input and employs the binary WSO algorithm for intelligent feature selection. Additionally, the LSTM classifier is utilized in the fully connected CNN layer for sample classification.

Experimental validation on four datasets demonstrates the efficacy of the proposed method. Notably, in the ISCX-URL-2016 dataset, it achieves impressive accuracy, precision, and sensitivity of 97.94, 97.82, and 97.76%, respectively. Similarly, in the PhishTank dataset, it achieves high accuracy, precision, and sensitivity of 96.78, 95.67, and 95.71%, respectively, outperforming traditional deep learning methods like LSTM, CNN, and CNN-LSTM.

Key advantages of the proposed method include:

- Dataset balancing with GAN deep learning
- Coding of URLs strings in the form of a numerical matrix and according to CNN input structure
- Extracting the features of URLs with deep learning based on traversing URL characters
- Intelligent selection of selected CNN features with WSO swarm intelligence
- Optimizing CNN output with WSO algorithm and optimal selection of features to reduce attack detection error
- High accuracy of the proposed model in detecting phishing attacks
- Ability to detect zero-day attacks
- Evaluation of the proposed method on four different datasets

However, The relatively long time required to train the model and the complexity encountered during the training phase represent significant challenges of the proposed method.

In future research, there is potential to further emphasize the exploration of advanced techniques to enhance the proposed method's efficacy. Specifically, the utilization of Apache Spark architecture as a distributed system holds promise for accelerating training processes and diminishing overall training time. Additionally, future endeavors can delve deeper into the integration of embedded learning within the fully connected CNN layer to elevate the performance of the method. These avenues of research are poised to significantly augment the effectiveness and efficiency of phishing attack detection methodologies, thereby advancing cybersecurity measures.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11760-024-03204-2.

Authors' contributions All authors have contributed equally to the manuscript.

Funding The authors received no financial support for the research, authorship, and/or publication of this article.

Data availability Dataset is publicly available at web link https://unb.ca/cic/datasets/url-2016.html.

Declarations

Conflict of Interests The authors have no conflicts of interest to disclose.

Ethical approval The study is classified as non-human subject research, and any permission is not required.

References

1. Zieni, R., Massari, L., Calzarossa, M.C.: Phishing or not phishing? A survey on the detection of phishing websites. IEEE Access **11**, 18499–18519 (2023)
2. Buckley, J., Lottridge, D., Murphy, J.G., Corballis, P.M.: Indicators of employee phishing email behaviours: intuition, elaboration, attention, and email typology. Int. J. Hum. Comput. Stud. **172**, 102996 (2023)
3. Anuar Mokhtar, A.H., et al.: A preliminary investigation on user factors of phishing E-mail. Central Asia Caucasus **23**(1), 55 (2022). https://doi.org/10.37178/ca-c.23.1.189
4. Siddiqi, M.A., Pak, W., Siddiqi, M.A.: A study on the psychology of social engineering-based cyberattacks and existing countermeasures. Appl. Sci. **12**(12), 6042 (2022)
5. Al-Qahtani, A.F., Cresci, S.: The COVID-19 scamdemic: a survey of phishing attacks and their countermeasures during COVID-19. IET Inf. Secur. **16**(5), 324–345 (2022)
6. Zahra, S.R., Chishti, M.A., Baba, A.I., Wu, F.: Detecting Covid-19 chaos driven phishing/malicious URL attacks by a fuzzy logic and data mining based intelligence system. Egypt. Inf. J. **23**(2), 197–214 (2022)
7. Wei, Y., Sekiya, Y.: Sufficiency of ensemble machine learning methods for phishing websites detection. IEEE Access **10**, 124103–124113 (2022)
8. Oest, A. et al.: {PhishTime}: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists. in *29th USENIX Security Symposium (USENIX Security 20)*, pp. 379–396, (2020)
9. da Silva, C.M.R., Feitosa, E.L., Garcia, V.C.: Heuristic-based strategy for Phishing prediction: a survey of URL-based approach. Comput. Secur. **88**, 101613 (2020)
10. Paturi, R., Swathi, L., Pavithra, K.S., Mounika, R., Alekhya, C.: Detection of phishing attacks using visual similarity model. in *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, IEEE, pp. 1355–1361, (2022)
11. Divakaran, D. M., Oest A.: Phishing detection leveraging machine learning and deep learning: a review. *arXiv preprint* arXiv:2205.07411, 2022.
12. Aljabri, M., Mirza S.: Phishing attacks detection using machine learning and deep learning models," in *2022 7th International Conference on Data Science and Machine Learning Applications (CDMA)*, IEEE, pp. 175–180, (2022)
13. Aljofey, A., et al.: An effective detection approach for phishing websites using URL and HTML features. Sci. Rep. **12**(1), 8842 (2022)
14. Raja, A. S., Pradeepa, G., Arulkumar N.: Mudhr: Malicious URL detection using heuristic rules based approach. in *AIP Conference Proceedings*, AIP Publishing, (2022)
15. Van Dooremaal, B., Burda, P., Allodi, L., Zannone, N.: Combining text and visual features to improve the identification of cloned webpages for early phishing detection. in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, pp. 1–10, (2021)
16. Wazirali, R., Yaghoubi, E., Abujazar, M.S.S., Ahmad, R., Vakili, A.H.: State-of-the-art review on energy and load forecasting in microgrids using artificial neural networks, machine learning, and deep learning techniques. Electric Power Syst. Res. **225**, 109792 (2023)
17. Xiao, X., Zhang, D., Hu, G., Jiang, Y., Xia, S.: CNN–MHSA: a convolutional neural network and multi-head self-attention combined approach for detecting phishing websites. Neural Netw. **125**, 303–312 (2020)
18. Zhu, E., Yuan, Q., Chen, Z., Li, X., Fang, X.: CCBLA: a lightweight phishing detection model based on CNN, BiLSTM, and attention mechanism. Cognit Comput **15**(4), 1320–1333 (2023)
19. Zhu, E., Ju, Y., Chen, Z., Liu, F., Fang, X.: DTOF-ANN: an artificial neural network phishing detection model based on decision tree and optimal features. Appl. Soft Comput. **95**, 106505 (2020)
20. Rambabu, V., Malathi, K., Mahaveerakannan, R.: An innovative method to predict the accuracy of phishing websites by comparing logistic regression algorithm with support vector machine algorithm. in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, IEEE, pp. 646–650, (2022)
21. Balogun, A.O., Mojeed, H.A., Adewole, K.S., Akintola, A.G., Salihu, S.A., Bajeh, A.O., Jimoh, R.G.: Optimized decision forest for website phishing detection. In: Silhavy, R., Silhavy, P., Prokopova, Z. (eds.) Data Science and Intelligent Systems: Proceedings of 5th Computational Methods in Systems and Software 2021, pp. 568–582. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-90321-3_47
22. Roy, S.S., Awad, A.I., Amare, L.A., Erkihun, M.T., Anas, M.: Multimodel phishing URL detection using LSTM, bidirectional LSTM, and GRU models. Future Internet **14**(11), 340 (2022)
23. Nepal, S., Gurung, H., Nepal R.: Phishing URL detection using CNN-LSTM and random forest classifier. (2022)
24. Rahman, A.U., Al-Obeidat, F., Tubaishat, A., Shah, B., Anwar, S., Halim, Z.: Discovering the correlation between phishing susceptibility causing data biases and big five personality traits using C-GAN. IEEE Trans. Comput. Soc. Syst. (2022). https://doi.org/10.1109/TCSS.2022.3201153
25. Hota, H.S., Sharma, D., Shrivas, A.: An integrated approach of proposed pruning based feature selection technique (PBFST) for phishing e-mail detection. Recent Adv. Comput. Sci. Commun. Formerly: Recent Patents Comput. Sci. **15**(5), 683–692 (2022)
26. Ahmed, D.S., Hussein, A.P.D.K.Q., Allah, H.A.A.A.: Phishing websites detection model based on decision tree algorithm and best feature selection method. Turkish J. Comput. Math. Edu. (TURCOMAT) **13**(1), 100–107 (2022)
27. Wang, J.: An improved genetic algorithm for web phishing detection feature selection. in *2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, IEEE, pp. 130–134, (2022)
28. Priya, S., Selvakumar, S., Velusamy, R.L.: PaSOFuAC: particle swarm optimization based fuzzy associative classifier for detecting phishing websites. Wirel. Pers. Commun. **125**(1), 755–784 (2022)
29. Shuaib, M., et al.: Whale optimization algorithm-based email spam feature selection method using rotation forest algorithm for classification. SN Appl. Sci. **1**, 1–17 (2019)
30. Sabahno, M., Safara, F.: ISHO: improved spotted hyena optimization algorithm for phishing website detection. Multimed. Tools Appl. **81**(24), 34677–34696 (2022)
31. Jayaraj, R., Pushpalatha, A., Sangeetha, K., Kamaleshwar, T., Shree, S.U., Damodaran, D.: Intrusion detection based on phishing detection with machine learning. Meas. Sens. **31**, 101003 (2024)
32. Zhu, E., Cheng, K., Zhang, Z., Wang, H.: PDHF: effective phishing detection model combining optimal artificial and automatic deep features. Comput. Secur. **136**, 103561 (2024)
33. Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q.E.U., Saleem, K., Faheem, M.H.: A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. Electronics **12**(1), 232 (2023)
34. Asiri, S., Xiao, Y., Alzahrani, S., Li, S., Li, T.: A survey of intelligent detection designs of HTML URL phishing attacks. IEEE Access **11**, 6421–6443 (2023)
35. Ahammad, S.K.H., et al.: Phishing URL detection using machine learning methods. Adv. Eng. Softw. **173**, 103288 (2022)

36. Braik, M., Hammouri, A., Atwan, J., Al-Betar, M.A., Awadallah, M.A.: White Shark Optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems. Knowl. Based Syst. **243**, 108457 (2022)
37. Guo, B., Zhang, Y., Xu, C., Shi, F., Li, Y., Zhang, M.: HinPhish: An effective phishing detection approach based on heterogeneous information networks. Appl. Sci. **11**(20), 9733 (2021)
38. Basit, A., Zafar, M., Liu, X., Javed, A.R., Jalil, Z., Kifayat, K.: A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommun. Syst. **76**, 139–154 (2021)
39. Hijji, M., Alam, G.: A multivocal literature review on growing social engineering based cyber-attacks/threats during the COVID-19 pandemic: challenges and prospective solutions. IEEE Access **9**, 7152–7169 (2021)
40. Daengsi, T., Pornpongtechavanich, P., Wuttidittachotti, P.: Cybersecurity awareness enhancement: a study of the effects of age and gender of Thai employees associated with phishing attacks. Educ. Inf. Technol. (Dordr) **1**, 1–24 (2022)
41. Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Rodriguez-Galan, G., Martínez-Cepeda, V., Nuñez-Agurto, D.: Comparative study of deep learning algorithms in the detection of phishing attacks based on HTML and text obtained from web pages. in *International Conference on Applied Technologies*, Springer, pp. 386–398, (2022)
42. Adane, K., Beyene, B.: Phishing website detection with and without proper feature selection techniques: Machine learning approach. in *The International Symposium on Computer Science, Digital Economy and Intelligent Systems*, Springer, pp. 745–756, (2022)
43. Wen, T., Xiao, Y., Wang, A., Wang, H.: A novel hybrid feature fusion model for detecting phishing scam on Ethereum using deep neural network. Expert Syst. Appl. **211**, 118463 (2023)
44. Tan, C.C.L., Chiew, K.L., Yong, K.S.C., Sebastian, Y., Than, J.C.M., Tiong, W.K.: Hybrid phishing detection using joint visual and textual identity. Expert Syst. Appl. **220**, 119723 (2023)
45. Bozkir, A.S., Dalgic, F.C., Aydos, M.: GramBeddings: a new neural network for URL based identification of phishing web pages through n-gram embeddings. Comput. Secur. **124**, 102964 (2023)
46. Shirazi, H., Muramudalige, S.R., Ray, I., Jayasumana, A.P., Wang, H.: Adversarial autoencoder data synthesis for enhancing machine learning-based phishing detection algorithms. IEEE Trans. Services Comput. **16**(4), 2411–2422 (2023). https://doi.org/10.1109/TSC.2023.3234806
47. Nordin, N.S., Ismail, M.A.: A hybridization of butterfly optimization algorithm and harmony search for fuzzy modelling in phishing attack detection. Neural Comput. Appl. **35**(7), 5501–5512 (2023)
48. Lin Z., et al.: A structured self-attentive sentence embedding," *arXiv preprint* arXiv:1703.03130, (2017)
49. Shieh, C.-S., et al.: Detection of adversarial DDoS attacks using generative adversarial networks with dual discriminators. Symmetry (Basel) **14**(1), 66 (2022)
50. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. *Adv Neural Inf Process Syst*, vol. 28, (2015)
51. Alshehri, M., Abugabah, A., Algarni, A., Almotairi, S.: Character-level word encoding deep learning model for combating cyber threats in phishing URL detection. Comput. Electr. Eng. **100**, 107868 (2022)
52. Makhadmeh, S.N., Al-Betar, M.A., Assaleh, K., Kassaymeh, S.: A hybrid white shark equilibrium optimizer for power scheduling problem based IoT. IEEE Access **10**, 132212–132231 (2022)
53. Oh, S., Yu, M., Cho, S., Noh, S., Chun, H.: Bi-LSTM-Augmented deep neural network for multi-Gbps VCSEL-based visible light communication link. Sensors **22**(11), 4145 (2022)
54. Bu, S.-J., Kim, H.-J.: Optimized URL feature selection based on genetic-algorithm-embedded deep learning for phishing website detection. Electronics (Basel) **11**(7), 1090 (2022)
55. Iuga, C., Nurse, J.R.C., Erola, A.: Baiting the hook: factors impacting susceptibility to phishing attacks. HCIS **6**, 1–20 (2016)
56. Le, H., Pham, Q., Sahoo, D., Hoi, S. C. H.: URLNet: Learning a URL representation with deep learning for malicious URL detection. *arXiv preprint* arXiv:1802.03162, (2018)
57. Tajaddodianfar, F., Stokes, J. W., Gururajan A.: Texception: a character/word-level deep learning model for phishing URL detection. in *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 2857–2861, (2020)
58. Prabakaran, M.K., Meenakshi Sundaram, P., Chandrasekar, A.D.: An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders. IET Inf. Secur. **17**(3), 423–440 (2023)