International Journal of Information Security
https://doi.org/10.1007/s10207-024-00851-x

REGULAR CONTRIBUTION

A cyber defense system against phishing attacks with deep learning game theory and LSTM-CNN with African vulture optimization algorithm (AVOA)

Mustafa Ahmed Elberri[1] · Ümit Tokeşer[2] · Javad Rahebi[3] · Jose Manuel Lopez-Guede[4]

Accepted: 8 April 2024

Abstract
Phishing attacks pose a significant threat to online security, utilizing fake websites to steal sensitive user information. Deep learning techniques, particularly convolutional neural networks (CNNs), have emerged as promising tools for detecting phishing attacks. However, traditional CNN-based image classification methods face limitations in effectively identifying fake pages. To address this challenge, we propose an image-based coding approach for detecting phishing attacks using a CNN-LSTM hybrid model. This approach combines SMOTE, an enhanced GAN based on the Autoencoder network, and swarm intelligence algorithms to balance the dataset, select informative features, and generate grayscale images. Experiments on three benchmark datasets demonstrate that the proposed method achieves superior accuracy, precision, and sensitivity compared to other techniques, effectively identifying phishing attacks and enhancing online security.

Keywords Fake pages · Phishing attacks · SMOTE · Deep learning · Game theory · Convolutional neural networks · LSTM · Feature selection · African vulture optimization algorithm (AVOA)

1 Introduction

The widespread adoption of online financial services, including those offered by PayPal, has increased the potential impact of ransomware attacks [1]. While WannaCry incorporated some phishing-like elements, it was primarily a ransomware attack that exploited a vulnerability in Windows systems, known as EternalBlue, to infect computers connected to the Internet without users' interaction.

Online services are not only for paying money but online shopping, such as shopping on Amazon, is also one of the valuable services of the Internet [2]. The number of Internet users has grown significantly in recent years, the reason for which is the wide variety of web pages. Web pages have made Internet users use them daily by providing services anywhere and anytime. Despite the countless uses and advantages of the Internet and web pages, each technology has its challenges. Fake web pages are one of the main challenges on the Internet that cause cyber-attacks [3].

Fake pages look very similar to legitimate or original pages. In fake pages, users' valuable information, such as usernames and passwords, is stolen [4]. Phishing attacks involve social engineering tactics, where individuals known as phishers or online criminals fraudulently acquire users' information. Phishers use fake sites to deceive users and send links to phishing pages to victims via email, social networks, or text message services [5]. Phishing attacks have a simple cycle and only require a little expertise. Phishing

✉ Jose Manuel Lopez-Guede
jm.lopez@ehu.eus

Mustafa Ahmed Elberri
albrry1@gmail.com

Ümit Tokeşer
utokeser@kastamonu.edu.tr

Javad Rahebi
cevatrahebi@topkapi.edu.tr

[1] Department of Material Science and Engineering, University of Kastamonua, 37150 Kastamonu, Turkey

[2] Department of Mathematics, University of Kastamonu, 37150 Kastamonu, Turkey

[3] Department of Software Engineering, Istanbul Topkapi University, 34087 Istanbul, Turkey

[4] Department of Systems and Automatic Control, Faculty of Engineering of Vitoria-Gasteiz, University of the Basque Country (UPV/EHU), C/Nieves Cano 12, 01006 Vitoria-Gasteiz, Spain

Published online: 05 May 2024

attacks are often used to steal bank information and customer bank account information. The effectiveness of a phishing attack relies on several factors, such as the resemblance of the website, the use of social engineering, and users' levels of knowledge [6]. There is no exact estimate of the number of phishing attacks, but reports from security agencies show that these attacks have grown significantly in recent years. Phishing attacks are not only limited to fake web pages; this challenge also exists in digital currencies. WannaCry is an example of a phishing attack in cryptocurrency that caused a loss worth 4 billion dollars. This cyber-attack affected more than 300,000 computers in 150 countries. Phishing constitutes more than half of all cybercrimes within the Ethereum [7].

The increased use of social networks and online financial platforms has significantly raised the vulnerability to cyber-attacks for both businesses and individuals. Spear phishing is one of the most critical types of phishing attacks. This type of attack sends a commercial and deceptive email to specific businesses and their users. According to estimates by the Federal Bureau of Investigation) (FBI), Phishing victims worldwide lost more than $26 billion to email-based phishing attacks between 2016 and 2019. In 2018, approximately $60 million was lost to internet users due to spear phishing attacks in Australia. The share of the United States (US) in phishing attacks was about 39%, the share of the United Kingdom (UK) was 26%, and the share of Australia was 11% in the years 2018–2019 [8].

Phishing attacks use human nature and communication to succeed. In social engineering, a phisher or thief communicates with users through tricks and deception. For example, a hacker can pretend to be a bank employee and send a deceptive email to users. If users need more knowledge, they will be tricked by the hacker and reveal their valuable information. The most basic method is to detect attacks and then deal with them to deal with phishing attacks. There are various methods to deal with phishing attacks, which are classified into two categories: user training and software methods [9].

Methods of informing users during the training process is a long and expensive process. Software methods for detecting phishing attacks are divided into list-based approaches [10], based on visual similarity [11], heuristic methods [12], based on machine learning [13], and deep learning [14]. Heuristic methods recognize web pages based on evidence such as the character type within the address, the character length, the number of address points, etc. Still, these methods have a significant error rate. Blacklist methods use pattern and address matching with the database of phishing addresses to detect attacks. Blacklist methods require a lot of memory and time to search. Unlike blacklist and heuristic approaches, machine and deep learning have the capability to identify new and zero-day attacks. Visual methods to detect attacks use image processing and visual elements such as images and logos [15]. Deep learning methods are more capable of pattern recognition than machine learning methods and offer a much greater degree of learning than machine learning methods. The challenge of using CNN architecture is in processing strings as input. URL addresses should be coded as images to train the CNN neural network.

The primary objective of this paper is to introduce a novel approach for identifying phishing attacks utilizing a deep learning architecture that combines CNN and LSTM, drawing insights from game theory. Another goal of the manuscript is to improve the CNN architecture in combination with swarm intelligence methods, including the AVOA. The authors' motivation to present a phishing attack detection method based on CNN neural network architecture is to reduce the error rate of phishing attack detection, practical use of CNN architecture in network security applications, and reduce the damage of attacks by timely detection of attacks. The suggested approach for detecting phishing attacks extracts the basic features of web pages and links. In the second step, the SMOTE method [16] is combined with deep learning based on GAN and Autoencoder to balance the number of phishing and legal classes and increase learning accuracy. The AVOA [17] selects the feature in the second step. The proposed method provides a binary adaptation of the African vulture algorithm with the aim of dimensionality reduction for the dataset. Dimensionality reduction causes the input of deep learning to reduce and the speed of learning to increase. Reducing the dimensions allows deep learning to focus on essential features and mitigate the error of the CNN neural network as a classifier. The proposed method converts the reduced-dimension data set into gray image format and trains CNN and LSTM deep learning architecture by images. The contribution of the authors of this research to detect phishing attacks is presented below:

- Data set balancing with SMOTE technique combined with deep learning based on GAN and Autoencoder network
- Introducing a binary variant of the AVOA with chaos theory
- Variable selection with swarm intelligence of the African vulture algorithm in detecting phishing attacks
- Improvement of CNN architecture with swarm intelligence in the detection of phishing attacks
- Integration of deep learning architecture of CNN with LSTM
- Presenting a new method for coding strings and numbers in the form of gray images as input to CNN and LSTM networks

The manuscript is prepared and edited in 5 parts: Section 2, research background, and related works in phishing reviews. Section 3 presents a suggested approach for detecting phishing attacks. Section 4 presents the analysis of the suggested

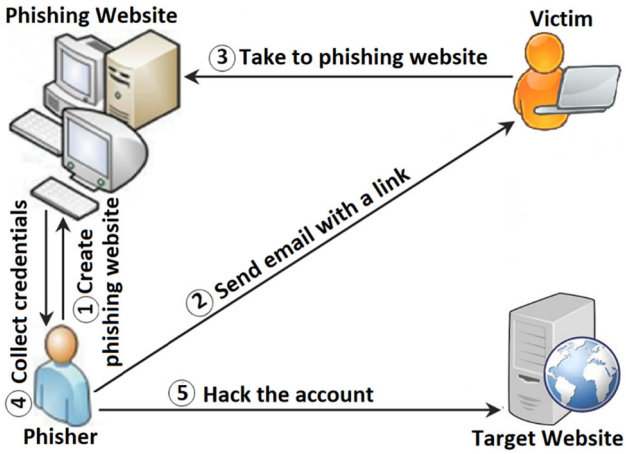

Fig. 1 The cycle of phishing attacks [18]

strategy for detecting phishing attacks in Python and MATLAB. Section 5 presents research results and future works to improve the detection performance of phishing attacks.

2 Related works

According to Fig. 1, phishing attacks have a multi-stage cycle. In the first step, the phisher spoofs an original website and creates a similar fake website. In the next step, the fake site is uploaded by a phisher to the Internet. The hacker sends links to fake pages to users through communication media to direct users or victims. If users click the fake site link, they enter the fake website, and the thief steals their information.

The thief uses social engineering methods, especially email, to deceive users. An example of fake pages on the Internet is shown in Fig. 2. In fake pages, the website's appearance resembles the original website. Still, analyzing the website's source code, including the number of absurd links, it recognized that it was phishing. In this example, several empty links on the web page indicate that the website is phishing. Fake sites have a set of features that prove they are phishing. In many phishing sites, JavaScript codes prevent users from right-clicking. Fake sites are usually short-lived, and their domains have been registered recently. Some features are related to web page links. For example, fake web pages contain @ characters or unusual port addresses. According to Fig. 2, as per the reports from the Anti-Phishing Working Group (APWG), it is evident that the total count of phishing websites in 2020 surpassed the figures for all four seasons in 2019. According to this report, 165,772 phishing sites in the first quarter of 2020 increased to about 637,302 in the fourth quarter of 2020.

Figure 3 illustrates various techniques employed for detecting phishing attacks. These methods include list-based, similarity-based, and machine learning approaches. Blacklist methods use a database including rules and attack signatures to detect phishing attacks. These methods are simple but require a lot of memory, and their search time is significant.

Visual similarity measurement methods use images and logos of web pages to analyze web pages. Visual methods for detecting phishing attacks require advanced image processing tools and algorithms. These methods are very complicated, and their error rate is not small.

Other methods, heuristic methods, use exploratory functions to detect phishing attacks. In these methods, discoveries such as the length of the website address, the number of subdomains, the lifetime of the domain, and the presence of unique characters, are checked. Heuristic methods have acceptable accuracy, but they cannot detect zero-day attacks. Machine learning and deep learning methods include numerous methods for detecting phishing attacks. Deep learning and machine learning methods offer the advantage of detecting zero-day attacks. In the rest of this section, several studies in detecting phishing attacks are examined and reviewed.

Advantages of deep learning and machine learning in detecting zero-day attacks

Conventional signature-based intrusion detection systems fail to detect zero-day attacks, while deep learning and machine learning offer a promising solution by adapting to new patterns without relying on signatures [21] and [22].

Advantages of Deep Learning and Machine Learning for Detecting Zero-Day Attacks [21] and [22]:

1. Pattern Recognition: Deep learning algorithms excel in recognizing intricate patterns crucial for identifying zero-day attacks that may display subtle or unusual behaviors.
2. Adaptability: Machine learning models can continuously learn and adapt to new data, enabling them to identify emerging threats and evolving attack patterns.
3. Feature Extraction: Deep learning models automatically extract relevant features from raw data, reducing the need for manual feature engineering and enhancing overall detection effectiveness.

Addressing Overfitting and False Positives [23]:

1. Despite their potential, deep learning and machine learning methods face challenges, including the overfitting problem leading to high false positives. Mitigation techniques include:
2. Data Augmentation: Increasing the quantity and diversity of training data aids the model in generalizing better to new data.
3. Regularization Techniques: Methods like dropout and early stopping prevent overfitting by penalizing complex models and avoiding excessive memorization.

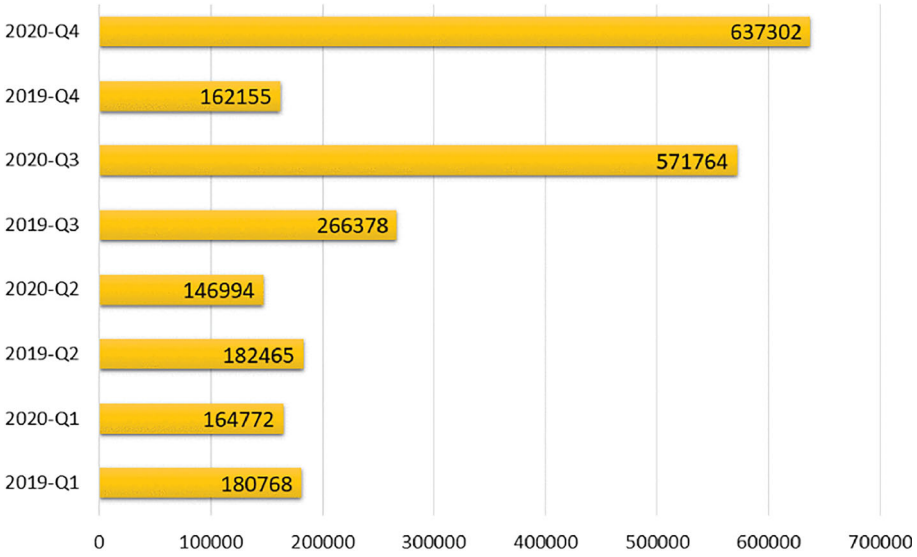

Fig. 2 Increase in the number of phishing attacks in 2019 and 2020 [19]

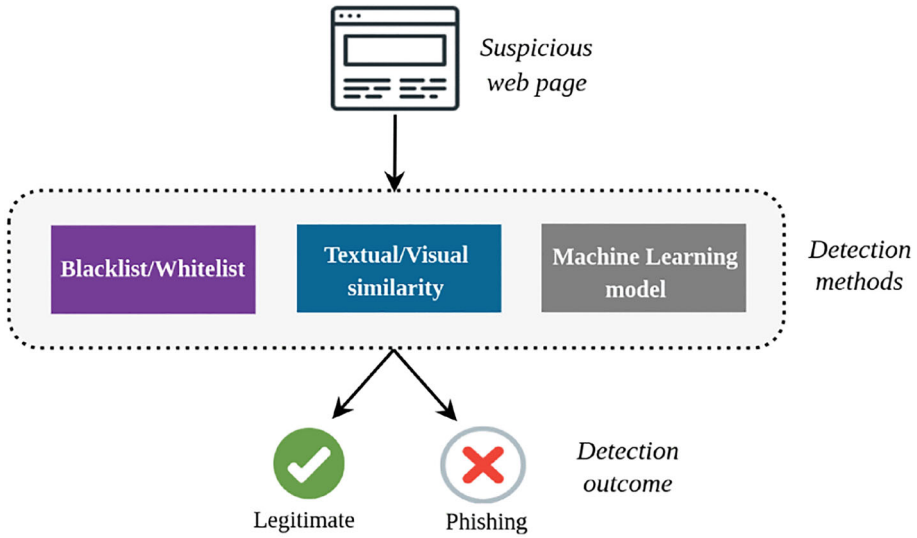

Fig. 3 Types of methods to deal with phishing attacks [20]

4. Ensemble Methods: Combining multiple models into an ensemble improves performance and reduces the risk of overfitting.
5. False positives are further addressed through post-processing techniques like anomaly detection and reputation-based filtering. Anomaly detection identifies deviations from expected patterns, while reputation-based filtering considers data source trustworthiness.

Combining Deep Learning and Traditional Techniques [24, 25] and [26]:

Deep learning and machine learning complement, rather than replace, traditional security techniques. A hybrid approach offers:

1. Complementary Strengths: Leveraging the strengths of both approaches to address their respective weaknesses.
2. Improved Accuracy: Reducing false positives and enhancing overall detection accuracy by combining different detection methods.
3. Enhanced Adaptability: The hybrid system adapts to changing threats by incorporating new signatures and rules into traditional components while continuously learning from new data in machine learning models.

In [27], phishing detection techniques are based on artificial intelligence and human behavior review. This study investigates automatic phishing detection techniques available on websites based on the combined methods of artificial intelligence and human behavior.

In [28], phishing websites are identified using machine learning. This research uses decision tree, random forest, and gradient boosting classifier (GBC) methods with three feature selection techniques to detect phishing attacks. They use

a dataset with 89 features. Through experiments, it has been demonstrated that the gradient-boosting classifier method exhibits greater accuracy in attack detection when compared to decision trees and random forests.

In [29], they presented an attribute selection method utilizing a particle swarm optimization (PSO) algorithm to identify phishing websites. Experimental results reveal that employing the PSO-based attribute selection model enhances the accuracy of machine learning models in detecting phishing attacks. Experimental findings demonstrate that the neural network achieves the highest level of accuracy in detecting phishing attacks, reaching a rate of 97.81%.

In [30], they presented an LSTM-based email phishing detection method. They proposed a phishing email detection framework that combines federated learning with LSTM. The results show that their method can achieve a prediction accuracy of 83%.

In [31], they propose the detection of phishing attacks based on cloud computing by combining deep learning models. They employ the LSTM model for URL analysis, utilize the YOLOv2 model for logo analysis, and apply the triple network model for visual similarity analysis. The results show that combining models is more accurate than individual models in detecting phishing attacks.

In [32], phishing attack detection is accomplished through the utilization of deep learning models and hyperparameter optimization. This research endeavors to advance deep learning models and optimize meta-parameters to achieve a high level of accuracy in detecting phishing websites. This research investigates three deep learning algorithm architectures, including recognition models founded on short-term memory, fully connected deep neural networks, and CNNs. The tests demonstrated that their suggested model achieved the highest accuracy, approximately 97.7%. The tests showed that the network search algorithm and the genetic algorithm increased the accuracy of the models by about 0.1% and 1%.

In [33], the presents a method of detecting phishing websites using a hybrid algorithm. In data preprocessing, an adaptive artificial sampling approach is used to deal with unbalanced data. In the second phase, Rao's meta-heuristic algorithm removes additional features. This research uses a KNN classifier to distinguish fake pages from real ones. The information is taken from the UCI Machine Learning Repository. Experiments show that the obtained classification accuracy is 97.44%.

In [34], they presented an approach to detect phishing pages on health-related websites using visual techniques. This study uses three classifiers: a decision tree, a random forest, and a support vector machine to detect phishing attacks. Experiments revealed that the decision tree achieves the greatest precision in detecting phishing attacks.

In [35], the presents an approach to detect phishing attacks in social network services with a convolutional neural network. Simple notification service (SNS) phishing is one representative social engineering attack that abuses people's feelings and trust. In these attacks, the attacker establishes a close emotional connection with the victims. In this study, they employ a CNNs as the name of the Telegram chatbot. Experiments demonstrated that the Text-CNN method achieves better accuracy than LSTM in detecting phishing attacks.

In [36], a deep learning method is accessible for identifying phishing websites through visual similarity. Phishing detection methods relying on visual similarity involve significant complexity in the extraction of visual features. This work uses a transfer learning technique to extract features as input to machine learning algorithms. The experimental findings indicate that combining VGG16 with a machine learning-based algorithm yields accurate results in detecting phishing attacks.

In [37], deep semantic fusion models are present for detecting phishing websites. This study thoroughly incorporates semantic information across various scales. The three suggested models, namely the Multi-scale Data-layer Fusion (MDF) model, Multi-scale Feature-layer Fusion (MFF) model, and Multi-scale In-depth Fusion (MIF) model, are employed for attack detection. Experimental results indicate that the MIF model excels in detecting phishing attacks on a complex dataset, boasting a low false positive rate of 0.0047. Experiments conducted in real-world settings demonstrate that the suggested model is both competitive and practical for real diagnostic scenarios.

In [38], phishing URL detection can be performed using a temporal convolutional network (TCN). They use a new deep learning technique, TCN with word embedding, to identify phishing URLs. Experimental results showed that their method detects phishing internet addresses with 98.95% and 98% accuracy and sensitivity, respectively.

In [39], detecting spam and fake e-mails is proposed in phishing attacks with a Bi-LSTM neural network. This study employs various machine learning and deep learning algorithms, including the random forest classifier, artificial neural network, support vector machine, long-term memory, short-term memory, and bidirectional LSTM. Evaluations indicate that deep learning techniques exhibit higher accuracy compared to machine learning methods.

In [40], an approach to generate synthetic URLs based on GAN is present to detect phishing URLs. Existing URL databases have a small number of samples and must be balanced. In this research, they proposed training a GAN network called WGAN-GP to generate malicious URLs from existing phishing URL data to solve this challenge. They used LSTM and GRU classifiers to detect phishing attacks.

In [41], a deep learning-based approach is available for the detection of zero-day phishing attacks. In this study, they introduced a method of integrating deep learning and logically programmed domain knowledge to detect phishing attacks. They proposed neural and logical classifiers in combination with the typical learning method. The tests showed that their method improves the sensitivity index by about 3%.

In the examination of URL analysis, the study outlined in [42] stands out. The researchers conscientiously crafted a resilient ensemble learning model, utilizing a multivariate filter-based approach for feature selection to identify potentially malicious URLs. Their approach, integrating correlation feature selection and statistical *t*-tests to pinpoint crucial features, yielded notable outcomes. The achieved accuracy rates were remarkable, reaching 97% in the initial dataset and an impressive 99.25% accuracy in the second dataset.

In the domain of identifying phishing websites, the investigation discussed in [43] provides valuable perspectives through a comparative analysis of logistic regression and random forest. The authors utilized correlation-based feature selection to improve the accuracy of classifying phishing websites. The results, demonstrating accuracy rates of 93.035% for logistic regression and 96.834% for random forest, emphasize the efficacy of their methodology. The subsequent feature selection stage further raised the accuracy rates to 92.718% and 97.015%, respectively.

In [44], they propose the detection of phishing websites through the application of natural language processing and a deep learning algorithm. The proposed work is to build an automatic and hybrid model using a random forest algorithm in machine learning with a convolutional neural network algorithm in deep learning, which is applied to identify and classify phishing in URL and web page content in an automatic machine. Their approach demonstrates higher accuracy in detecting phishing attacks compared to standard machine-learning strategies.

Table 1 compares the reviewed works in phishing detection. The analysis of related studies and works shows that in most studies, deep learning and machine learning are used to detect attacks. Visual methods for detecting phishing attacks have many limitations, and visual algorithms need a lot of time to analyze website images. If the machine learning methods lack feature selection, they have average accuracy. Data set balancing methods are essential for increasing the accuracy of machine learning and deep learning methods. Most studies focus on URL addresses, but for detecting phishing attacks, the characteristics of content, domain, search engines, and source code of web pages also have valuable information. The proposed method provides an approach based on deep learning with swarm intelligence and data set balancing to solve these challenges. Examining related works to detect phishing attacks shows that there are the following challenges:

- Deep learning methods have low accuracy if trained on unbalanced data.
- Deep learning methods without dimensionality reduction of the input take a long time to train.
- Failure to use optimization methods in deep learning reduces the accuracy of detecting phishing attacks.
- In many studies, simple GA and PSO algorithms use to detect phishing attacks, but these approaches cannot accurately search the problem space. Swarm intelligence algorithms presented recently have more complexity and robust modeling than previous metaheuristic algorithms, such as GA and PSO.
- Failure to use chaos theory in swarm intelligence behaviors will reduce global search and their accuracy and metaheuristic algorithms trapped in local optima.
- Failure to integrate deep learning architectures such as CNN and LSTM reduces the accuracy of detecting phishing attacks.

3 Methodology

The suggested approach to identifying phishing attacks addresses many of the difficulties encountered in previous research significantly. Examining related works showed that some studies used deep learning, such as GAN, to balance the dataset. In previous studies, the SMOTE method is applied to fix dataset imbalances. The suggested method combines an improved version of GAN with SMOTE to make the data set more balanced. In contrast to earlier research, the proposed method employs swarm intelligence for feature selection and identifying fundamental features when combining deep learning techniques. The AVOA has been presented recently and has high accuracy in calculating optimal solutions. In this paper, swarm intelligence is used to improve deep learning performance, including CNN and LSTM, for the purpose of detecting phishing attacks. The innovation of this manuscript includes the following:

- Balancing the data set with the GAN version of the probabilistic type
- Improvement of the GAN model with SMOTE method
- Feature extraction with the N-Gram method
- Improvement of AVOA with chaos theory
- Introducing a binary adaptation of the AVOA
- Conducting feature selection using the AVOA
- Providing an optimal approach and a combination of CNN and LSTM in detecting fake pages and links

Table 1 Method, advantages, and disadvantages of related works

Research	Year	Method	Advantages	Disadvantages	Existing approaches
[27]	2023	Artificial intelligence and human behavior	Using human behavior	The behavior of the phisher has not been investigated	Requires additional data collection and analysis
[28]	2023	Decision tree, random forest and gradient boosting classifier	Check out lots of features	Lack of intelligent feature selection	May increase computational complexity
[29]	2023	PSO-based feature selection	Accuracy is about 97.81%	Unbalanced data set	May require additional parameter tuning
[30]	2022	LSTM based phishing email detection	Detection of zero-day attacks	low accuracy	May have higher computational cost
[31]	2022	Cloud computing by combining deep learning models	Analysis of content and visual appearance	High complexity	Requires significant infrastructure and expertise
[32]	2022	Deep learning and optimization of deep learning hyperparameters	More accurate than CNN and LSTM	high overhead	May require more time and resources for optimization
[33]	2022	Combined algorithm of Rao and KNN	Accuracy 97.44%	KNN is a weak classifier	May require more complex data preprocessing
[34]	2023	Visual techniques and machine learning	Appropriate precision with decision tree	Time complexity	May provide less interpretable results
[35]	2022	Text-CNN	More accurate than LSTM	Unbalanced	May require additional training data
[36]	2022	Deep learning and visual similarity	VGG is highly accurate	Non-use of content	May have higher computational cost
[37]	2022	Deep semantic fusion	False positive rate 0.0047	High time overhead	Mayrequire more complex model architecture
[38]	2022	Temporal convolutional neural network	98.95% accuracy	A lot of training time	May require more data for effective training
[39]	2023	Bi-LSTM	More accurate than LSTM	Unbalanced data set	May have higher computational cost
[40]	2021	GAN network for phishing URL detection	Balancing the dataset	Failure to use content features	May require more complex data preprocessing
[41]	2021	Deep learning and programmed knowledge	Improving the sensitivity index by about 3%	Lack of feature selection and lack of balancing of the data set	May require more time and resources for optimization
[42]	2023	Sustainable group learning	High accuracy	Lack of content analysis	May have higher computational cost
[43]	2023	Correlation-based feature selection	Reducing dimensions and reducing learning time	Unbalanced data set and low accuracy	May require more data for effective training
[44]	2023	Natural language processing and deep learning algorithms	High accuracy	No feature selection	May have higher computational cost

3.1 Proposed framework

The schematic representation of the proposed methodology in this paper, denoted as GS-A-CL (a combination of GAN and SMOTE with AVOA and CNN-LSTM), is illustrated in Fig. 4. The proposed approach for detecting phishing attacks consists of the following steps:

- Balancing the dataset through the integration of SMOTE and GAN methods
- Feature extraction with the N-Gram method
- Normalization of the data set
- Improving the performance of the African vulture algorithm with chaos theory
- Feature selection with improved African vulture algorithm
- Coding of selected features to color images

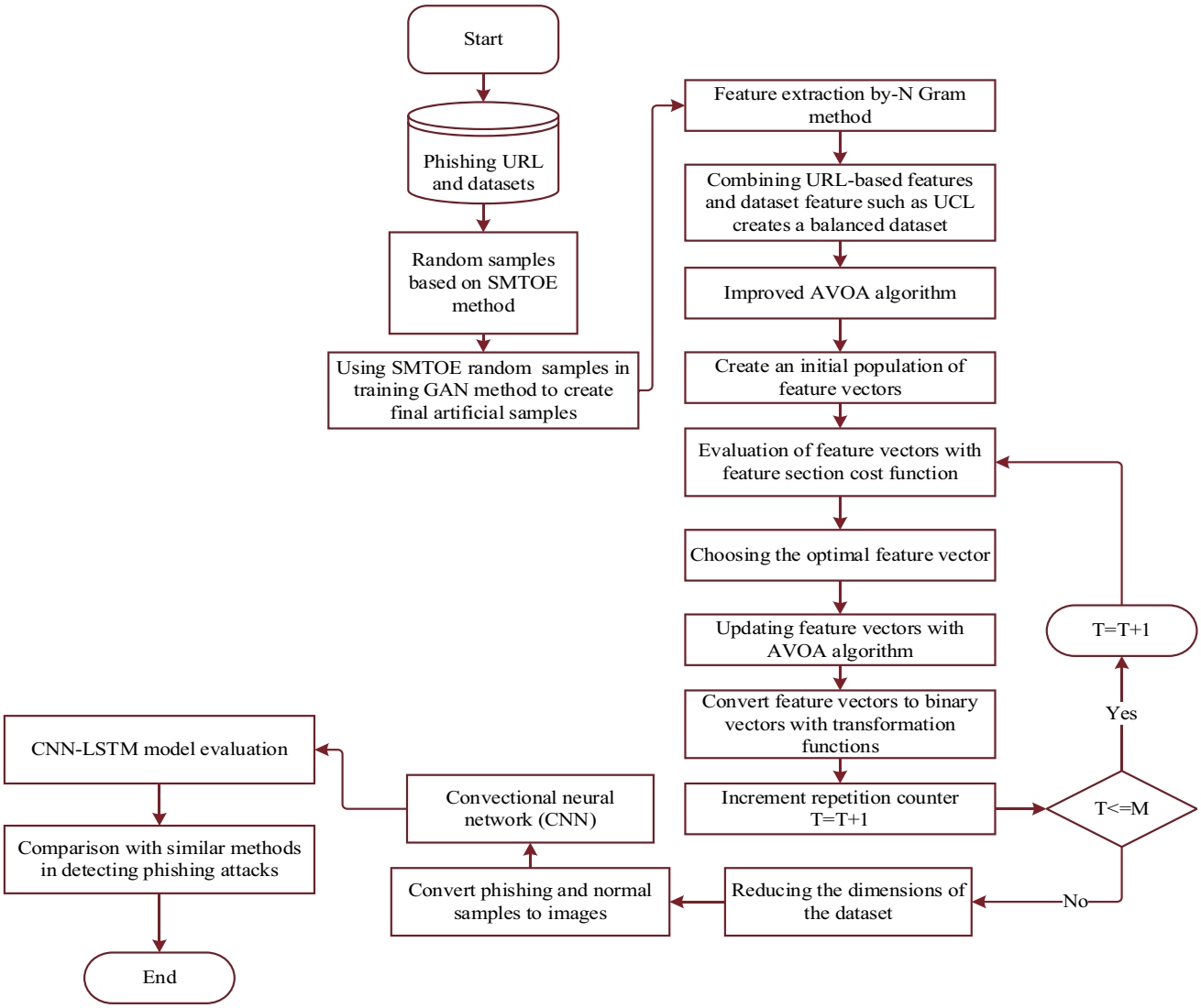

Fig. 4 Framework of the proposed method

- CNN neural network training combined with LSTM neural network
- Evaluation of the proposed method with test samples

3.1.1 Understanding the flowchart for phishing attack detection

This flowchart outlines a deep learning-based method for effectively detecting and identifying phishing attacks.

- Step 1: Data Augmentation
  - 1.1. Employ Synthetic Minority Over-Sampling Technique (SMOTE) to increase the number of minority class (phishing) samples in the dataset.
  - 1.2. Utilize an improved Generative Adversarial Network (GAN) based on the Autoencoder network to generate synthetic phishing samples.
- Step 2: Feature Extraction
  - 2.1. Apply the African Vulture Optimization Algorithm (AVOA), a swarm intelligence algorithm, to extract relevant information from phishing pages and fundamental features.
  - 2.2. Select crucial features from the extracted information.
- Step 3: Feature Representation
  - 3.1. Convert the selected features into grayscale images.
- Step 4: Deep Learning Model Training

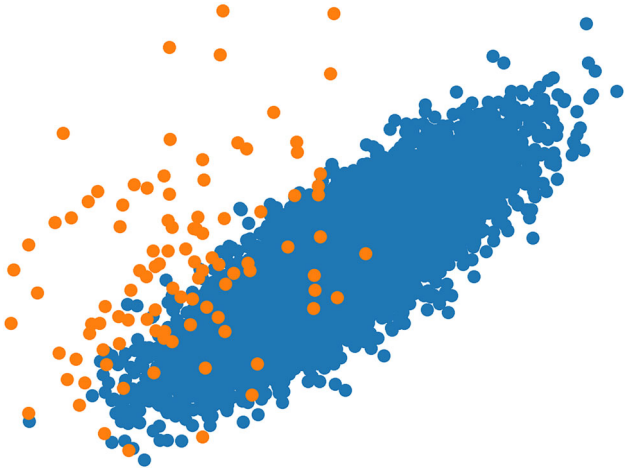

Fig. 5 Imbalanced data set with two classes

4.1. Train a convolutional neural network (CNN) in conjunction with Long Short-Term Memory (LSTM) using the grayscale images.

- Step 5: Evaluation

5.1. Evaluate the trained CNN-LSTM model on benchmark datasets to assess its performance in detecting phishing attacks.

Finding and Result: A novel deep learning approach employing SMOTE, an enhanced GAN, AVOA, and CNN-LSTM effectively detects phishing attacks, surpassing alternative methods in feature selection and achieving high accuracy, sensitivity, and precision across test datasets. This method holds promise for strengthening online security.

3.2 Dataset

Phishing URLs are used in datasets such as Phishtank and datasets such as UCI to train and evaluate the proposed method. The URL-based dataset is often unbalanced, and the UCI dataset has fewer phishing classes. In the proposed method, the feature extraction phase is performed on the URL, and the extracted features combine with the features of the UCI dataset. In addition to various URL features, the new data set has content, address, source code, domain, and search engine features.

3.3 Balancing the data set

A significant inequality in the number of samples between different classes characterizes an imbalanced dataset. In Fig. 5, an unbalanced data set is displayed. The unbalanced data set in machine learning and deep learning causes the classifier model to have a significant error.

Figure 5 shows two classes in blue and orange. The number of orange samples is far less than the blue class, the majority class. The samples in orange color have a smaller number and form the minority class. An efficient method to balance the data set is to add to the number of minority samples with an artificial data generation method. The GAN and SMOTE methods are used in most research to create random samples.

The GAN networks operate on a foundation of game theory, where a deep learning network referred to as a generator engages in competition with an adversarial process involving a discriminator. The discriminator distinguishes the samples generated from the generative network from the original data. The generator's role is to create fake samples and use its efforts to deceive the discriminator. The discriminator is a classifier that distinguishes fake from real samples. The game between the generator and the differentiator continues; the generator tries to deceive the differentiator, and the differentiator tries not to be deceived by the generator. The structure of the GAN network for generating random samples is shown in Fig. 6.

GANs produce samples that closely resemble the data in the training dataset, and their capacity to learn features is further amplified when combined with autoencoders. The variable autoencoder generates novel sample types that incorporate characteristics found in the training dataset. GAN exhibits instability during the learning phase, whereas the Variational Autoencoder (VAE) is comparatively more stable than GAN. The combination of GAN and VAE results in a generative model characterized by both high quality and stability. Utilizing synthetic data generated by a Variational Autoencoder Generative Adversarial Network (VAE-GAN) mitigates imbalanced data issues in the phishing attack detection model. The proposed method for generating random samples also uses the SMOTE algorithm to generate artificial samples as a generator input in GAN to increase the generator's ability to improve the discriminator's deception. In Fig. 7, the structure of the proposed SMOTE-VAE-GAN model for generating artificial examples of the minority class is displayed.

In the SMOTE method, random points are selected from the minority classes, and then, as shown in Fig. 8, the K numbers of their nearest neighbors are calculated, and their combination is produced as an output to obtain new samples. In Fig. 8, the majority samples show solid squares, and the minority samples show black circles. Red points are made from the connection between minority samples to increase the number of artificial samples. Figure 9 depicts the procedure for creating random and synthetic samples in the SMOTE method to achieve dataset balance.

Figure 9 displays the primary data as blue and green samples. The SMOTE method creates several artificial samples based on the nearest neighbor method from the minority

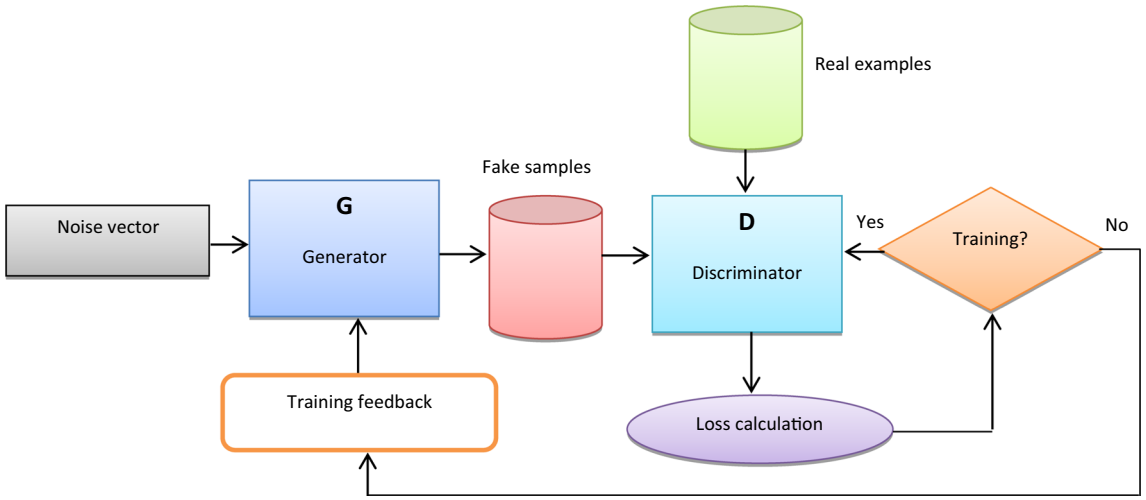

Fig. 6 Creating artificial samples with GAN network

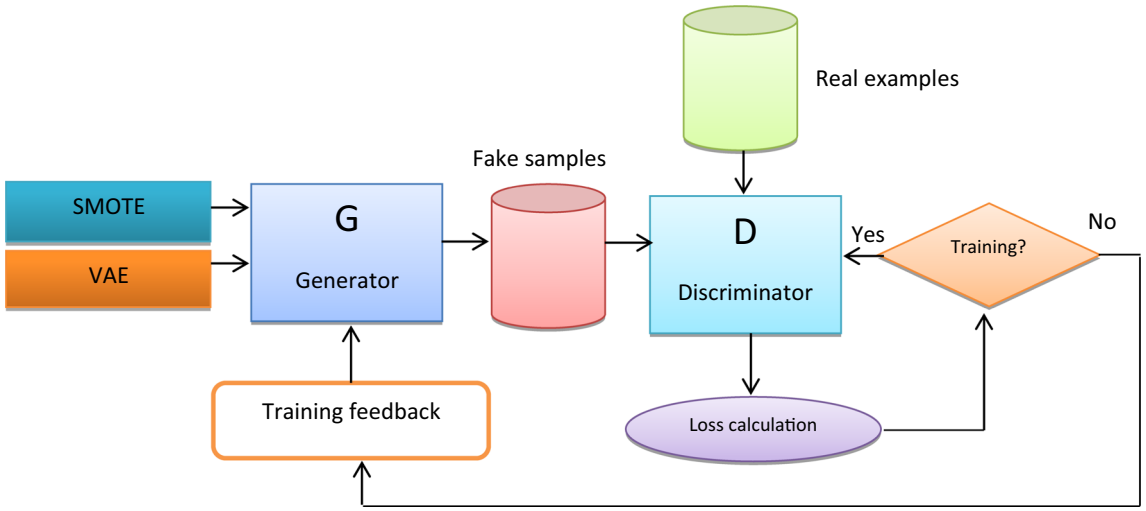

Fig. 7 Creation of artificial samples with SMOTE-VAE-GAN network

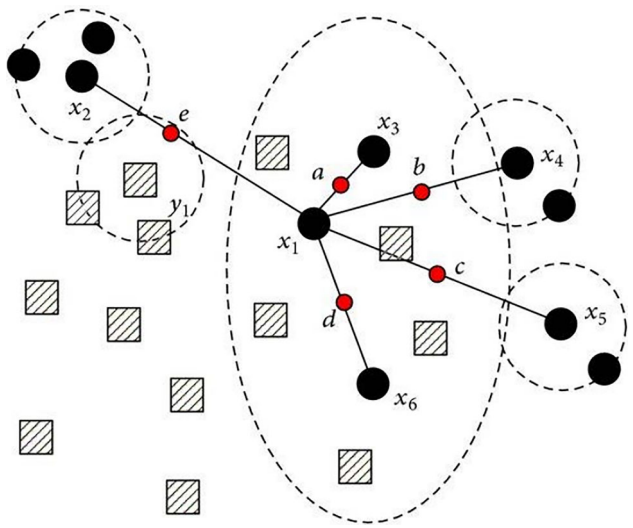

Fig. 8 Creation of synthetic samples by the SMOTE method [45]

samples or green circles, which are displayed in red circles. Increasing the quantity of synthetic samples leads to dataset balance. The pseudo-code of the SMOTE algorithm for balancing the data set is shown in Fig. 10.

In the proposed method, VAE-GAN is used to reduce the problem of unbalanced data. The GAN network exhibits instability during learning, whereas VAE generates more diverse examples and maintains relative stability throughout learning. Combining several generative models can provide the advantages of each model to create artificial samples. The VAE network consists of an encoder and a decoder module. The encoder transforms the input into a hidden vector, and subsequently, the decoder reconstructs this hidden vector into an approximate input. The encoder and decoder are shown as Eqs. (1) and (2) [47]:

$$z \sim \mathrm{Enc}(x) = q_\phi(z \mid x) \tag{1}$$

Fig. 9 Balancing the data set by the SMOTE method[46]

Fig. 10 SMOTE method Pseudocode

Algorithm: SMOTE algorithm Input: Xminor, Npentat, K

Output: $\mathbf{X_{synthetic\_samples}}$

Function SMOTE ($\mathbf{X_{Minor}}$, $\mathbf{X_{Percent}}$, **K)**

$\mathbf{X_{synthetic\_samples}}$**={}**

for i =1 to Size($\mathbf{X_{Minor}}$**) do**

M =K Nearest Neighbors ($\mathbf{X_i}$, $\mathbf{X_{Minor}}$ **k)**

Per= [Npercent /100]

while per! = 0 do

$\mathbf{X_{neighbour}}$ **= select random (M)**

$\mathbf{X_{synthetic\_samples}}$ **=**$\mathbf{X_i}$**+rand(0,1)*|** $\mathbf{X_{neighbour}}$ **-** $\mathbf{X_i}$ **|**

Per= Per-1

end while

end for

return $\mathbf{X_{synthetic\_samples}}$

$$\widehat{x} \sim \mathrm{Dec}(z) = p_\theta(x \mid z) \tag{2}$$

where x, z, and $\widehat{x}$ are input, hidden vector, and approximate input, respectively. φ and θ are the parameters of the encoder and decoder models. $q_\phi(z \mid x)$ is an approximation of $p_\theta(x \mid z)$. The VAE loss function as the sum of the reconstruction error according to Eq. (3) is [47]:

$$\begin{aligned} J_{VAE} &= J_{\text{recon}} + J_{\text{prior}}, \text{ with} \\ J_{\text{recon}} &= -\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] \\ J_{\text{prior}} &= D_{KL}\left(q_\phi(x \mid z) \| p_\theta(z)\right) \end{aligned} \tag{3}$$

The GAN model consists of a generator and a discriminator where DKL and $p_\theta(z)$ represent the Kullback–Leibler divergence and the prior distribution of z. The generator maps the hidden vector to the data space and assigns the discriminator the probability v and 1-v. The loss function of GAN with binary cross entropy according to generator and differentiator is shown in Eqs. (4) and (5) [47]:

$$v = \mathrm{Dis}(u) \in [0,\ 1],\ u = \mathrm{Gen}(w) \tag{4}$$

$$J_{GAN} = \log(\mathrm{Dis}(u)) + \log(1 - \mathrm{Dis}(\mathrm{Gen}(w))) \tag{5}$$

where u is an actual sample, and w is a random variable with probability function p(w). The objective and loss function of VAE-GAN formulate in Eq. (6) [47]:

$$\begin{aligned} J_{VAE-GAN} &= J_{prior} + J_{Dis_l} + J_{GAN}, \text{ with} \\ J_{Dis_l} &= -\mathbb{E}_{q(z|x)}\left[\log p(Dis_l(x) \mid z)\right], \end{aligned} \tag{6}$$

3.4 Pre-processing

In most cases, the values of the features of a data set have diverse and different values, which hurts machine learning. A practical way to increase the accuracy of machine learning is to transform all feature values of the dataset between zero and one. For normalization, the Min–Max method is used in the proposed method. The normalization equation for phishing attacks' data set considers Eq. (7).

$$x\prime = \frac{x - \min(x)}{max(x) - \min(x)} \tag{7}$$

x is the unnormalized value, and $x\prime$ is the normalized value of this feature. In the context of normalization, "min(x)" and "max(x)" represent the minimum and maximum values of the features within the phishing dataset, respectively.

3.5 Feature extraction and feature selection

For feature extraction, the Ngram method is used to extract URL features; the details of this technique are explained in [48]. The extracted features are URL-related and combined with data set features such as UCI. The feature selection phase is performed in the next step to select the basic features. In the suggested approach, every feature vector consists of elements that are either zero or one. If feature number*i* of a feature vector is equal to zero, feature number i is not selected in this data set, and otherwise, if it is equal to one, it is selected. In the suggested approach, each feature vector is a row of the equation matrix (8), and each column is the features associated with that feature vector. This matrix has n feature vectors or vultures; each vulture has d components. If the UCI data set is selected, because this data set has 30 features, this matrix is displayed like Eq. (9). In these equations, P is the initial population of vultures or feature vectors in detecting phishing attacks.

$$P = \begin{bmatrix} p_{1,1} & \cdots & \cdots & p_{1,j} & p_{1,d-1} & p_{1,d} \\ p_{2,1} & \cdots & \cdots & p_{2,j} & p_{2,d-1} & p_{2,d} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{n-1,1} & \cdots & \cdots & \cdots & p_{n-1,d-1} & p_{n-1,d} \\ p_{n,1} & \cdots & \cdots & p_{n,j} & p_{n,d-1} & p_{n,d} \end{bmatrix} \tag{8}$$

$$P = \begin{bmatrix} p_{1,1} & \cdots & \cdots & p_{1,j} & p_{1,29} & p_{1,30} \\ p_{2,1} & \cdots & \cdots & p_{2,j} & p_{2,29} & p_{2,30} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{n-1,1} & \cdots & \cdots & \cdots & p_{n-1,29} & p_{n-1,30} \\ p_{n,1} & \cdots & \cdots & p_{n,j} & p_{n,29} & p_{n,30} \end{bmatrix} \tag{9}$$

In the proposed method, an objective function needs to evaluate feature vectors, and it is possible to use Eq. (10) to evaluate a feature vector:

$$Cost(P_i) = \alpha.\frac{1}{m}.\sum_{i=1}^{m}\left(\overline{Y}_i - Y_i\right)^2 + \beta.\frac{\|P_i\|}{\|d\|} \tag{10}$$

In this Equation, the index is $\frac{1}{m}.\sum_{i=1}^{m}\left(\overline{Y}_i - Y_i\right)^2$ is the detection error of phishing attacks, and $\frac{\|P_i\|}{\|d\|}$ is equal to the amount of reducing the dimensions of the collection have given. These two components are used with alpha and beta weight coefficients, whose sum is usually equal to one, and alpha is chosen randomly, and beta is determined in terms of alpha. The objective function formulated as Eq. (11) made:

$$Cost(P_i) = \alpha.\frac{1}{m}.\sum_{i=1}^{m}\left(\overline{Y}_i - Y_i\right)^2 + (1-\alpha).\frac{\|P_i\|}{\|d\|} \tag{11}$$

In this equation and the objective function, Y_i and $\overline{Y}_i$ are the actual class value and the predicted value of the page type, respectively. Phishing and legal pages are coded with 1 and 0, respectively. $\|P_i\|$ is the number of selected features in the feature vector P_i, and ‖d‖ is the number of columns or the number of primary features of a data set. m in this equation is the number of samples available to evaluate phishing attacks in the proposed method. Typically, any feature vector capable of minimizing the value of this objective function is regarded as optimal. The AVOA algorithm has defaults, some of which are mentioned below:

- N solutions or vultures seek the best solution or prey in each stage.
- During every iteration of the algorithm, the population of vultures is divided into two groups. The worthiest vultures and the second worthy vultures are in the first category, and other vultures are in the other category, which plays the role of weak vultures.
- Each group has a different strategy for finding food, and their ability to find food is different.
- In the AVOA algorithm, it is assumed that hungry vultures have less energy to search and fly. For this reason, they try to move toward the prey and conflict with two optimal vultures near the food, and here, a kind of local search or productivity can be seen.
- Vultures that are not hungry have less aggressive behavior and are further away from the prey and cover and search most of the problem space, and here, a global and exploratory search is used.

Vulture's algorithm has four basic steps; the first step divides the population into strong and weak groups. If there

are n vultures, n-2 numbers are in the group of weak vultures, and the remaining two numbers are considered as two optimal vultures of the population. In the first step, the initial random population is created, and the most optimal and the second optimal solutions are selected as the most suitable solutions. In the vulture algorithm, two optimal vultures are selected as the head of the pack, and other vultures assume their position as an estimate of food and fly towards them.

The probability that a vulture-like X_i will move towards one of the two optimal vultures is considered in Eq. (12), based on the roulette wheel mechanism, or for simplicity, the probability of moving towards any worthy vulture set as 50% considered [17]:

$$p_i = \frac{F_i}{\sum_{i=1}^{n} F_i} \tag{12}$$

In this equation, F_i is the fitness of a vulture-like P_i, and p_i is the probability of a vulture moving towards one of the two optimal vultures of the population. The vulture algorithm assumes that optimal vultures are less hungry because they have more access to food are less aggressive, and fly more. Hungry vultures make more effort to move towards the optimal solution and are more aggressive and ready to fight for food. These vultures randomly choose and move in line with one of the two suitable solutions. Equations (13) and (14) are used to model the behavior of increasing hunger and decreasing satiety of vultures [17]:

$$t = \mathrm{h} \times \left(sin^{w}\left(\frac{\pi}{2} \times \frac{iter}{MaxIter}\right) + \cos\left(\frac{\pi}{2} \times \frac{iter}{MaxIter}\right) - 1 \right) \tag{13}$$

$$F = (2 \times \mathrm{rand} + 1) \times \mathrm{z} \times \left(1 - \frac{iter}{MaxIter}\right) + \mathrm{t} \tag{14}$$

In other words, the goal is that in the vulture algorithm, the solutions will stop searching globally and search more locally over time. This issue in the Vulture algorithm means that the vultures must fly towards and fight with two optimal vultures. The F function represents the vultures' satiety. It has a decreasing routine according to the algorithm's repetition, which means that the population sends toward the optimal solutions for local search in the last iterations. In these equations, iter is the iteration number of the AVOA algorithm, and MaxIter is the maximum iteration number of the AVOA algorithm. In these relationships, z is a random variable between $[-1, +1]$. In these equations, h is a uniform random number between $[-2, +2]$. w is a parameter that is used to generate t and F charts.

In these equations, t is a random step generation function for flight, and its value is generated between -0.5 and $+0.5$. The value of F based on t and the number of iterations of the Vulture algorithm has a decreasing routine. The AVOA algorithm determines the first and second optimal solutions in each iteration, identified as $\mathrm{BestVulture}_1$ and $\mathrm{BestVulture}_2$, respectively. In the suggested approach, based on the value of F, the type of search for finding food or attacking food is considered. In the first case, if $|F| > 1$, the vulture is not hungry; it is just looking for food and is flying, performing a global search. If $|F| \leq 1$, the vultures are hungry and are attacking the prey. In this case, a local or global search is performed. For the movement and flight of vultures, there are three parameters between zero and one, which consider p1, p2, and p3, and are equal to 0.6, 0.4, and 0.6, respectively.

In the global search mode, the vultures are complete and in expressions $|F| > 1$, and the vultures can roam in the problem space. Here, a random number is created for each vulture, and if it is smaller than p1, Eq. (15) is used, and if it is more significant than p1, Eq. (16) is used [17]:

$$P(i+1) = R(i) - D(i) \times F \tag{15}$$

$$D(i) = |X \times R(i) - P(i)| \tag{16}$$

In these equations, P(i) represents the current location of a vulture and P(i + 1) is the new location of a vulture, X is a random coefficient between 0 and 2, and R(i) is the random location of one of the vultures, including $\mathrm{BestVulture}_1$ and $\mathrm{BestVulture}_2$ is considered and i is the repetition counter of the algorithm (Eq. (17)) [17]:

$$P(i+1) = R(i) - F + rand((ub - lb).rand + lb) \tag{17}$$

The terms "ub" and "lb" represent the upper and lower bounds of the feature space or optimization problem, respectively. If the vultures are hungry, then $|F| \leq 1$, and in this case, two situations will occur, and it has shown some local search or productivity. If $|F| \geq 0.5$, then there are two states according to the variable p2, and if the random number is less than p2, Eq. (17) is used, and if it is more significant than Eq. (18), it is used [17]:

$$P(i+1) = D(i) \times (F + rand) - d(t) \tag{18}$$

$$d(t) = R(i) - P(i) \tag{19}$$

In these equations, D(i) is calculated like Eq. (16), and d(t) indicates the distance of a vulture from one of the two optimal vultures, $\mathrm{BestVulture}_1$ or $\mathrm{BestVulture}_2$. In other words, when many vultures gather together, one of the food sources can cause severe conflict over food preparation. In such cases, vultures with high physical strength prefer not to allow others to move toward their food. In this case, the hungry vulture tries to fly toward the other food, the side of the second worthy vulture, as shown in Fig. 11.

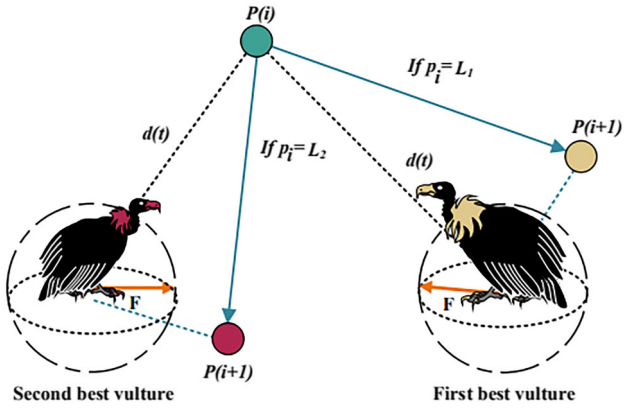

Fig. 11 Moving toward one of the two worthy vultures of the population [17]

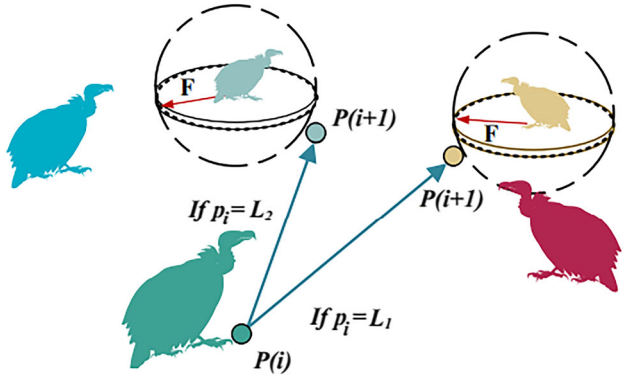

Fig. 12 Aggressive movement of the vulture toward the food position [17]

If |F|≥ 0.5 and the random number is more significant than p2, Eqs. (20), (21), and (22) are used to update the solutions [17]:

$$S_1 = R(i) \times \left(\frac{rand \times P(i)}{2\pi}\right) \times Cos(P(i)) \tag{20}$$

$$S_2 = R(i) \times \left(\frac{rand \times P(i)}{2\pi}\right) \times \sin(P(i)) \tag{21}$$

$$P(i+1) = R(i) - \frac{S_1 + S_2}{2} \tag{22}$$

If |F|< 0.5 and the random number is less than p3, Eqs. (23), (24) and (25) are used to update the solutions [17]:

$$A_1 = \text{BestVulture}_1(i) - \left(\frac{\text{BestVulture}_1 \times P(i)}{\text{BestVulture}_1 - P(i)^2}\right) \times F \tag{23}$$

$$A_2 = \text{BestVulture}_2(i) - \left(\frac{\text{BestVulture}_2 \times P(i)}{\text{BestVulture}_2 - P(i)^2}\right) \times F \tag{24}$$

$$P(i+1) = \frac{(A_1 + A_2)}{2} \tag{25}$$

If |F|< 0.5 and the random number is more significant than p3, Eq. (26) is used to update the solutions [17]:

$$P(i+1) = R(i) - |d(t)| \times F \times Levy(d) \tag{26}$$

In this equation, Levy(d) is a random flight function for d components and |d(t)| denotes the absolute value, which corresponds to the distance of a vulture from one of the optimal vulture. Figure 12 shows the aggressive behavior of a vulture moving toward prey or the optimal answer to the search radius F. As much as the repetition of the algorithm increases, the value of F decreases, and the vulture searches more around the optimal solution [17].

In Figure 13, the flowchart illustrating the proposed method for detecting phishing attacks is displayed.

One of the things that can improve the accuracy of metaheuristic algorithms is the use of random behavior based on chaos theory, which creates a sequence of random numbers that are used for each iteration of the algorithm for a random variable. In Eq. (27), the Logistic random and chaotic functions are introduced. Here n is the iteration number of the proposed algorithm.

$$X_{n+1} = a.X_n(1 - X_n) \tag{27}$$

In this function, X_0 is a number in the interval (0,1), but it cannot be the numbers 0, 0.25, 0.5, 0.75, and 1, and on the other hand, a is set equal to 2. In Eq. (28), the Tent random and chaotic function is introduced. Here n is the iteration number of the proposed algorithm.

$$X_{n+1} = \begin{cases} X_n/0.7 \; X_n < 0.7 \\ 10/3X_n(1 - X_n) \; otherwise \end{cases} \tag{28}$$

In this function, X_0 is a number in the interval (0,1), but it cannot be the numbers 0, 0.25, 0.5, 0.75, and 1, and on the other hand, a is set equal to 4. In equation (29), Sinusoidal random and chaotic functions are introduced. Here n is the iteration number of the proposed algorithm.

$$X_{n+1} = ax_n^2 \sin(\pi X_n) \tag{29}$$

In this function, X_0 is equal to 0.7, and a is equal to 2.3. In the proposed method, each of these chaotic maps is used to improve the AVOA and optimize the parameters of the AVOA with the help of these functions.

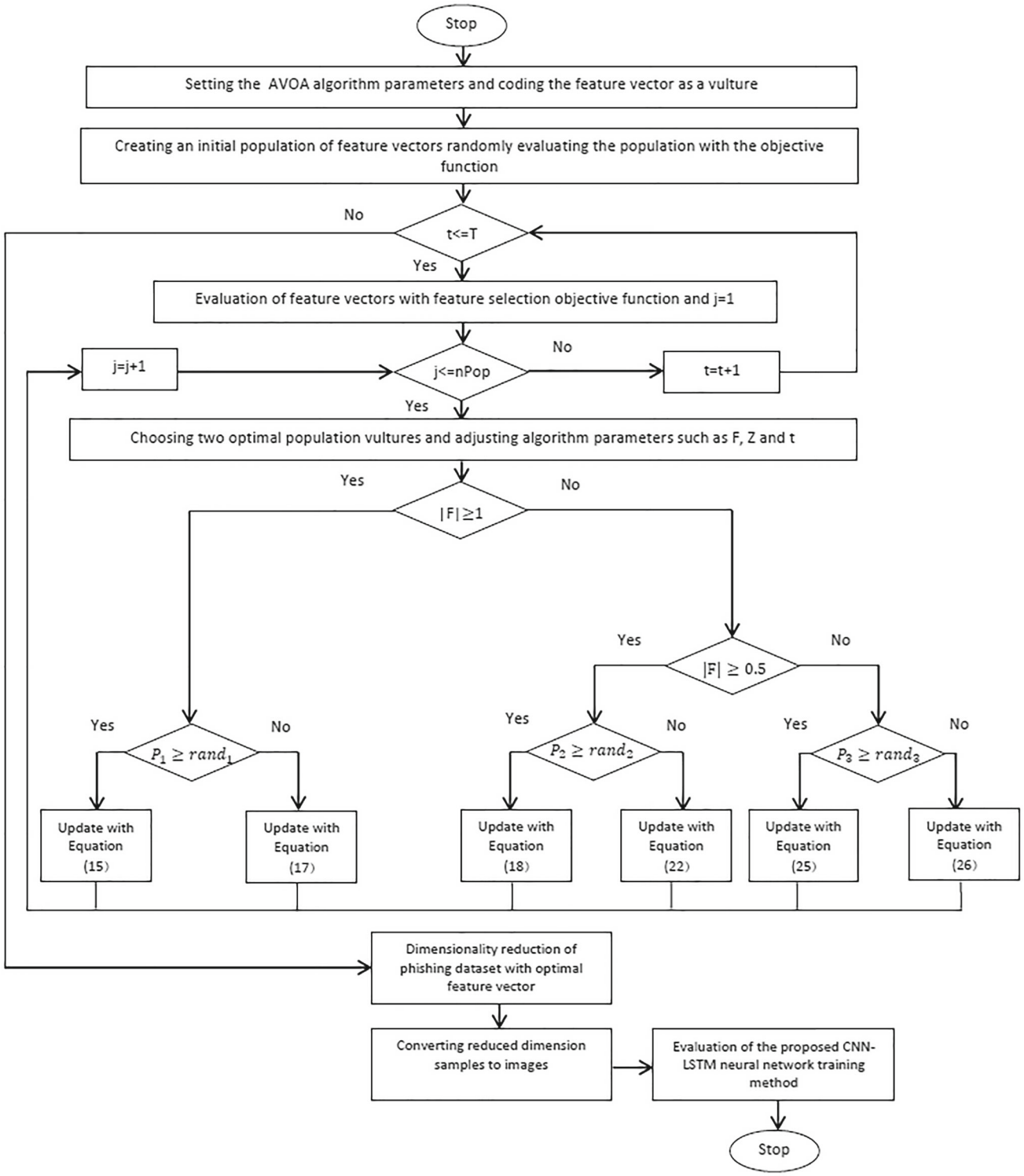

Fig. 13 Proposed feature selection flowchart

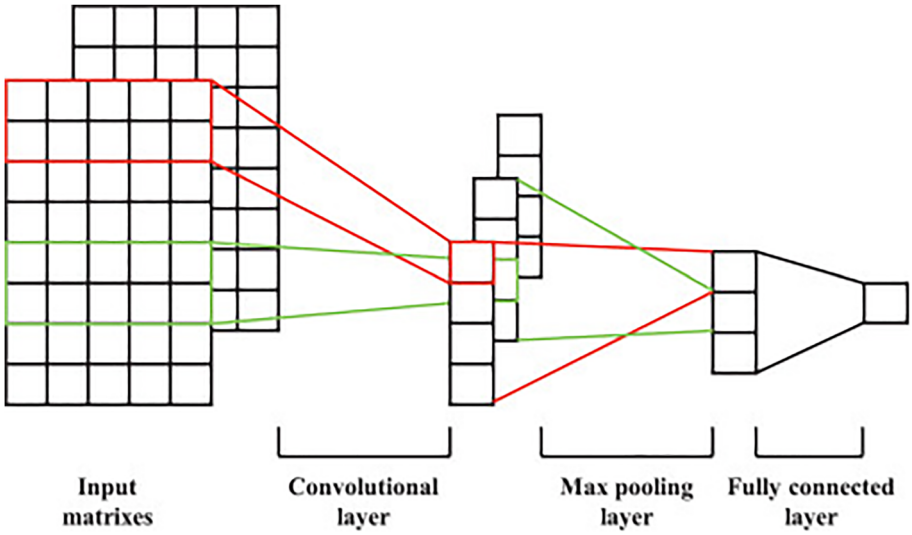

Fig. 14 Convolution neural network structure [13]

3.6 Classification with CNN-LSTM network

The selected features are applied to the dataset to reduce its dimensionality. Suppose the feature selection phase selects k features, then m phishing samples are selected from the data set, and a matrix of m in k creates a gray image. The numbers in the desired matrix are normalized between 0 and 255 to create a grayscale image. Gray images are created similarly for non-phishing samples and used to train the combined CNN-LSTM model. In the proposed method, the dimensions of the 224 × 224 images are set according to research [44]. In this case, there is a 224 × 244 matrix whose 224-k columns are empty and set to zero. The value of m, or the number of rows of the image matrix, is set to 224. In contrast to deep learning architectures, a CNN network is a specialized type of multilayer perceptron neural network, as a basic neural network is limited in its ability to learn complex features. CNNs excel in numerous applications, including image classification and medical image analysis. The proposed method selects images related to phishing and legal samples as input to the CNN neural network. CNN includes convolution, integration, and fully connected layers, as shown in Fig. 14.

LSTM is a variant of the RNN deep learning architecture designed specifically for tasks like time series analysis and classification. LSTM effectively uses a gating mechanism to deal with vanishing gradient problems in the training process. The LSTM memory cell has four gates named forgetting f, input gate i, control gate c, and output gate o. The fundamental configuration of the LSTM cell is presented in Fig. 15, and it consists of the output of the previous memory cell Ct-1 [49].

This neural network uses components such as the input signal at each time step Xt, the current memory cell Ct output, the previously hidden unit Ht − 1, and the currently hidden unit Ht. The forget gate determines the way in which the contribution from the previous time step is incorporated, resulting in a value ranging from zero to one for each data point in Ct-1. The input gate regulates the amount of input that is stored in the memory cell from the current time step. Meanwhile, the control gate updates the memory cell contents from Ct-1 to Ct.

The output gate dictates the extent to which the internal state influences the external state at the current time step. The symbol ⊗ represents the element-wise multiplication of vector elements, while ⊕ signifies the summation of vectors along with the application of the σ (sigma) function. To formulate the LSTM artificial neural network, Eqs. (30), (31), (32), (33), and (34) are used [49]:

$$f_t = \sigma(W_f.X_t + U_f.h_{t-1} + b_f) \tag{30}$$

$$O_t = \sigma(W_o.X_t + U_o.h_{t-1} + b_0) \tag{31}$$

$$\widetilde{C}_t = \tanh(W_c.X_t + U_c.h_{t-1} + b_c) \tag{32}$$

$$C_t = f_t.C_{t-1} + i_t.\widetilde{C}_t \tag{33}$$

$$h_t = \tanh(C_t) + O_t \tag{34}$$

The suggested approach uses the CNN neural network for feature analysis and the LSTM network for classifying phishing and legal samples. The structure of the neural network for detecting phishing samples is shown in Table 2.

4 Experimental results

This section implements the proposed approach for detecting phishing attacks in Python and MATLAB. The balancing and

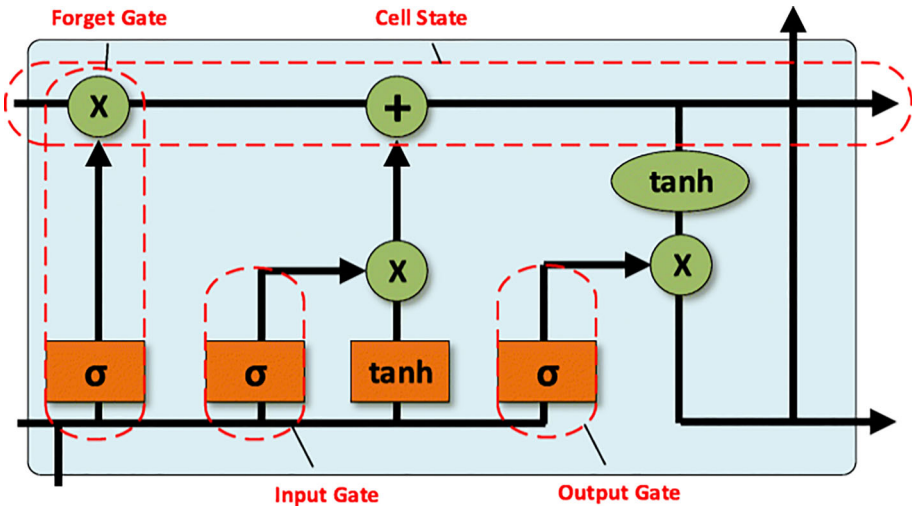

Fig. 15 Structure of an LSTM neural network cell

Table 2 CNN and LSTM hybrid neural network settings

Layer	Type	Kernel Size	Stride	Kernel	Input Size
1	Convolution2D	3 × 3	1	64	224 × 224 × 3
2	Convolution2D	3 × 3	1	64	224 × 224 × 64
3	Pool	2 × 2	2	–	224 × 224 × 64
4	Convolution2D	3 × 3	1	128	112 × 112 × 64
5	Convolution2D	3 × 3	1	128	112 × 112 × 128
6	Pool	2 × 2	2	–	112 × 112 × 128
7	Convolution2D	3 × 3	1	256	56 × 56 × 128
8	Convolution2D	3 × 3	1	256	56 × 56 × 256
9	Pool	2 × 2	2	–	56 × 56 × 256
10	Convolution2D	3 × 3	1	512	28 × 28 × 256
11	Convolution2D	3 × 3	1	512	28 × 28 × 512
12	Convolution2D	3 × 3	1	512	28 × 28 × 512
13	Pool	2 × 2	2	–	28 × 28 × 512
14	Convolution2D	3 × 3	1	512	14 × 14 × 512
15	Convolution2D	3 × 3	1	512	14 × 14 × 512
16	Convolution2D	3 × 3	1	512	14 × 14 × 512
17	Pool	2 × 2	2	–	14 × 14 × 512
18	LSTM	–	–	–	49 × 512
19	FC	–	–	64	25,088
20	Output	–	–	3	64

deep learning phase is implemented in Python, and the feature selection phase is implemented in MATLAB 2021.

4.1 Data set

In this research, the set of URLs is collected from the Phishtank database, and the UCI and Tan datasets are used to complete the tests. The proposed method balances the number of phishing and legal URLs then this dataset is employed for training in CNN-LSTM.

4.2 Evaluation metrics

Equations (35), (36), and (37) are used to formulate evaluation metrics such as accuracy, sensitivity, and precision.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{35}$$

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{36}$$

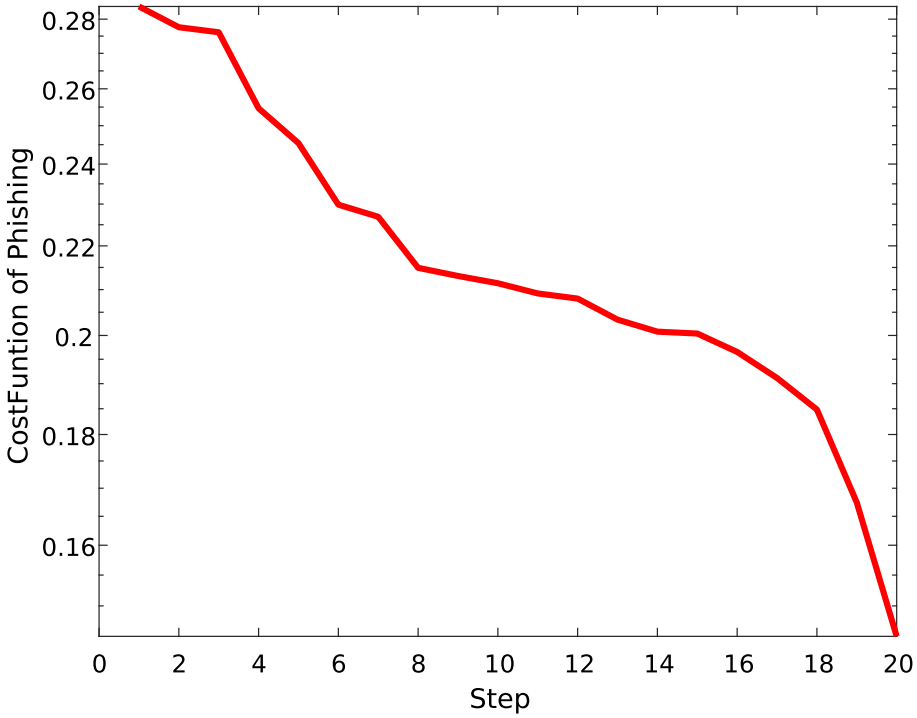

Fig. 16 Decreasing the objective function in the feature selection phase according to the iteration of the feature selection algorithm

$$Precision = \frac{TP}{TP + FP} \tag{37}$$

To calculate accuracy, sensitivity, and precision, the parameters TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives) are defined as follows:

- TP: It is a phishing sample, and the proposed method classifies the sample in the phishing category.
- FP: The sample is legal, and the proposed method incorrectly classifies the sample in the phishing category.
- TN: The sample is legal, and the proposed method classifies the sample in the legal category.
- FN: This is a phishing sample, and the proposed method incorrectly classifies the sample in the legal category.

4.3 Analysis of experiments

In this section, some of the tests are displayed. An essential phase in the proposed method is feature selection. Selecting the feature reduces the problem space in dimensions and the basic features used for CNN-LSTM training. The process of altering the value of the objective function during the feature selection phase and the classification error is illustrated in Figs. 16 and 17, respectively. The tests performed in the selection phase show the characteristics. The analysis of experiments shows that the suggested approach reduces the objective function of feature selection in terms of repetition of the AVOA. The decrease in the value of the feature selection objective function as the feature selection algorithm is repeated indicates:

- By optimizing the optimal feature vector, the average error of detecting phishing attacks is decreasing. Conversely, the accuracy in detecting attacks increases with each iteration of the meta-heuristic algorithm.
- The decrease in the error rate for attack detection results from the optimization of feature vectors. Furthermore, enhancing the feature vectors through optimization will lead to an improved effectiveness of the proposed method in detecting attacks.

4.4 Evaluation and results

The proposed method, or GS-A-CL, to detect phishing attacks uses three chaotic functions, Logistic, Tent, and Sinusoidal, for the random parameters of the AVOA. Figures 18, 19, and 20 show the precision, sensitivity rate, and correctness index of the suggested approach in the Phishtank, UCI, and Tan datasets, respectively.

The analysis in the PhishTank collection shows that if the Logistic chaotic function is used in the suggested approach, then the precision and sensitivity are higher than the Tent and Sinusoidal chaotic functions. If the Sinusoidal chaotic function is used, the accuracy will be 99.11%. In other words,

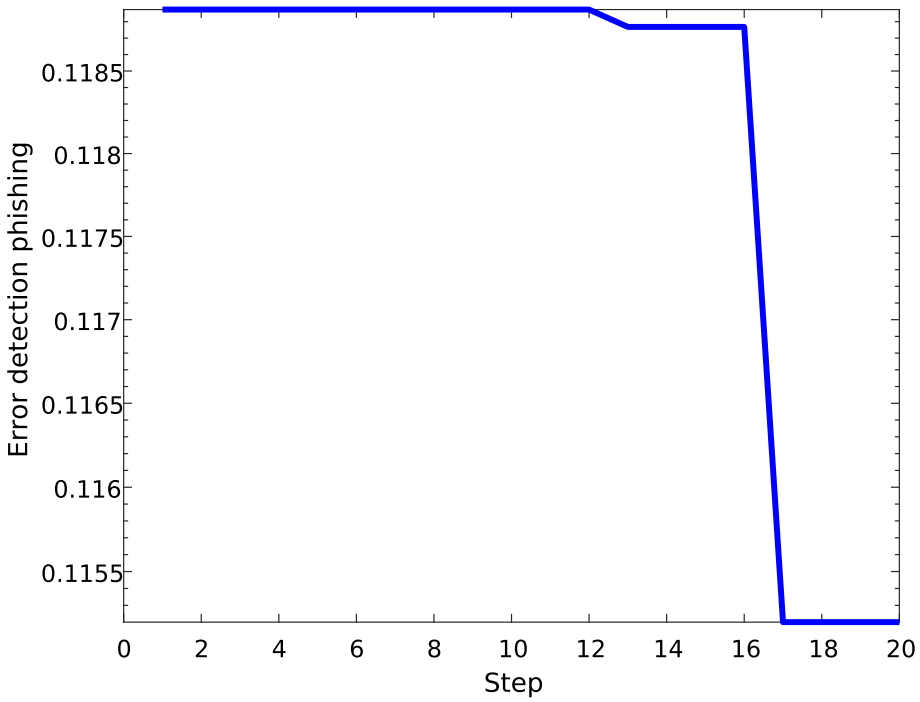

Fig. 17 Reducing the classification error in the feature selection phase by repeating the feature selection algorithm

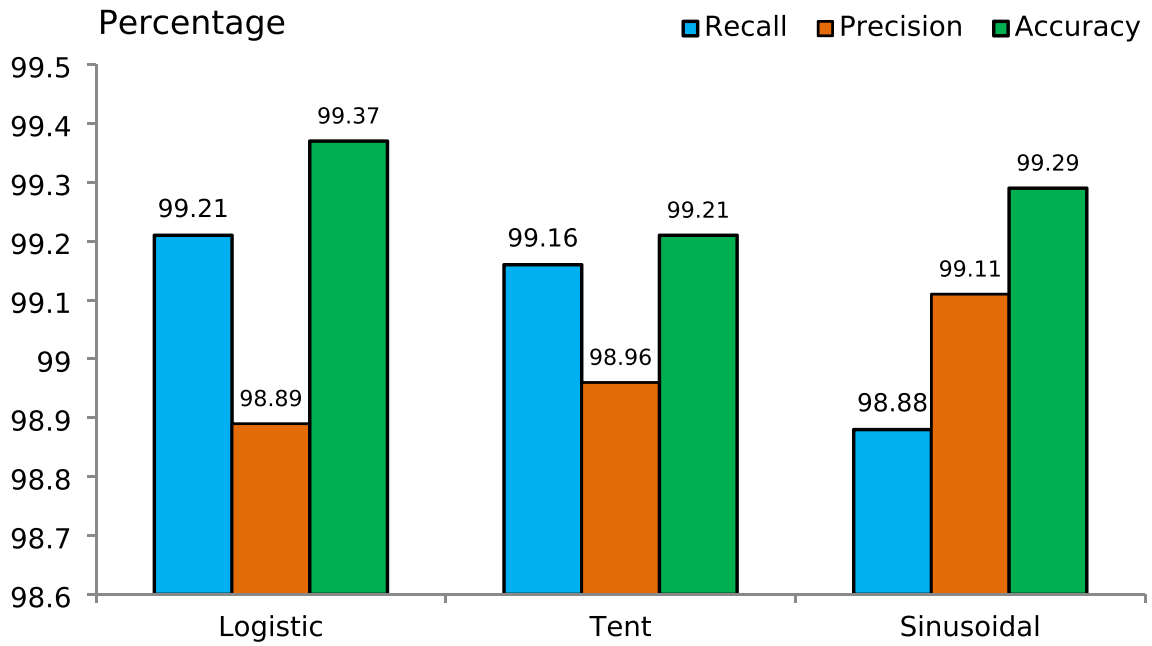

Fig. 18 Analysis of the evaluation metrics of the proposed method in the PhishTank dataset

the accuracy, precision, and sensitivity of the PhishTank dataset in the optimal state are equal to 99.37%, 99.11%, and 99.21%. In the UCI data set, the value of the accuracy index in the Logistic chaotic function is the maximum and is equal to 98.87%. The index of sensitivity and accuracy in the chaotic Tent function is maximum and equal to 98.93% and 98.54%, respectively. According to the Logistic chaotic function in the Tan data set, the accuracy, precision, and maximum sensitivity index are 98.79%, 98.64%, and 98.35%, respectively. Experiments show that in the general case, the Logistic Chaotic function performs better for detecting phishing attacks in terms of accuracy, sensitivity, and precision.

The suggested approach in the Tan dataset compares with the research results [32] in 2022, according to Table 3. The Tan dataset consists of 49 attributes, 48 serving as inputs,

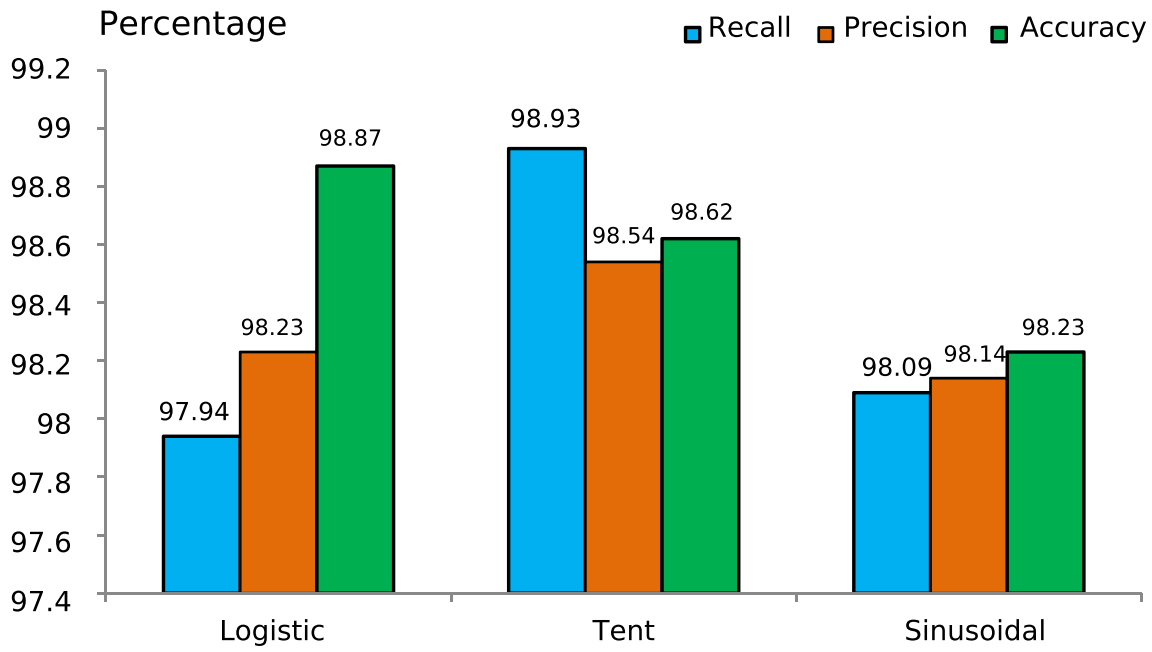

Fig. 19 Analysis of the evaluation metrics of the proposed method in the UCI dataset

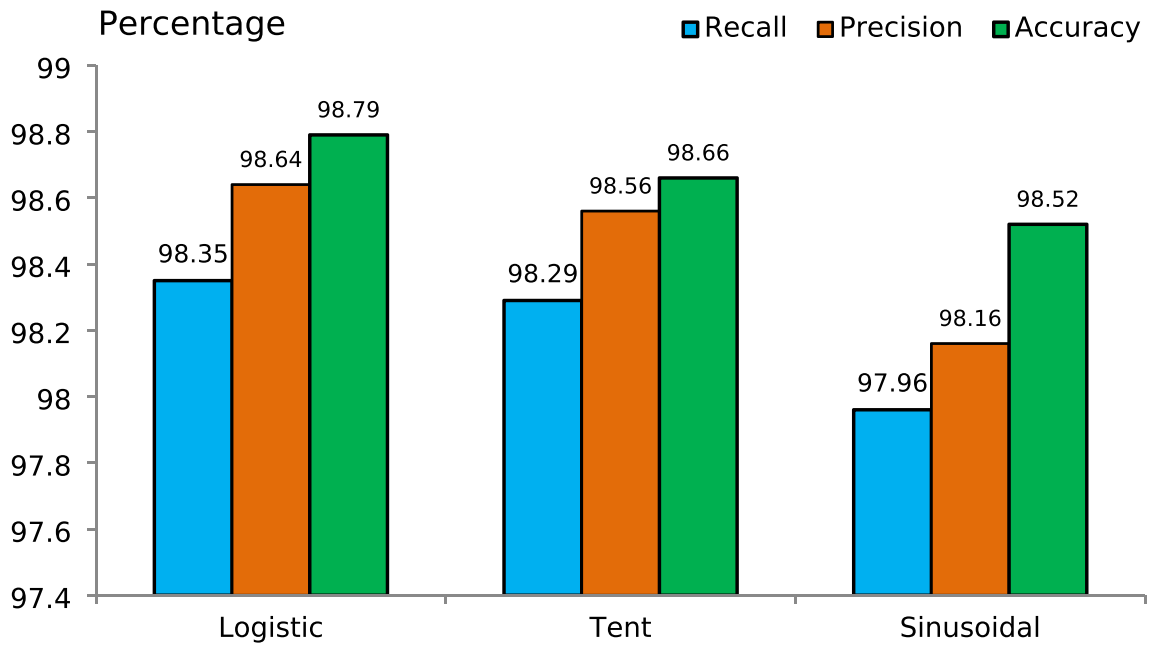

Fig. 20 Analysis of the evaluation metrics of the proposed method in the Tan dataset

while the 49th attribute functions as the output. In this dataset, the proposed method achieves an accuracy of 98.79%, a precision of 98.64%, and a sensitivity index of 98.35%. The proposed method demonstrates superior accuracy in detecting phishing attacks compared to LSTM-URL, LSTM-content, LSTM-all, FCnet-URL, FCnet-content, FCnet-all, CNN-URL, CNN-content, and CNN-all techniques.

The accuracy of the proposed method in the UCI dataset is evaluated in comparison to meta-heuristic methods, including the GOA, WSA, ABC, BWO, and GWO algorithms, as documented in reference [50]. The comparisons reveal that the accuracy scores for the GOA, GWO, BWO, WSA, and ABC algorithms are 98.43%, 98.33%, 98.27%, 98.67%, and 98.86%, respectively, as depicted in Fig. 21. The suggested

Table 3 Comparison of the proposed method with deep learning methods on the Tan dataset

Method	Accuracy	Precision	Sensitivity
LSTM-URL	90.40	91.45	89.13
LSTM-content	96.00	94.71	97.93
LSTM-all	97.37	96.71	98.07
FCnet-URL	90.47	89.41	91.80
FCnet-content	96.00	96.52	96.20
FCnet-all	96.77	96.36	97.20
CNN-URL	90.70	87.85	94.47
CNN-content	96.43	96.47	96.40
CNN-all	97.27	96.64	97.93
GS-A-CL	98.79	98.64	98.35

approach exceeds these meta-heuristic methods in detecting phishing attacks, with an accuracy rate of 98.89%. Among meta-heuristic algorithms, the bee optimization method has an accuracy of 98.88%, and its accuracy is slightly lower than the suggested approach.

The proposed method is more accurate on unbalanced data sets, but the Tan data set is balanced. Therefore, the balancing phase of the suggested method is relatively minor in the results and their improvement. The results for evaluating the suggested method in the Phistank dataset are compared with various deep learning-based research findings, as depicted in Fig. 22.

The diagram of Fig. 22 compares the proposed method with RNN-GRU, TransferLearning, Autoencoder, LSTM, and CNN-LSTM methods. The accuracy of RNN-GRU, TransferLearning, Autoencoder, LSTM, and CNN-LSTM methods is 99.18%, 97%, 97.82%, 99.57%, and 98.86%, respectively. The suggested method exhibits lower accuracy compared to the research presented in [51], but it still demonstrates superior performance in detecting attacks when compared to other deep learning methods. The proposed method is slightly weaker than the research [52] in detecting phishing URLs but is more accurate than the LSTM method in the UCI and Tan datasets. Table 4 compares the proposed method for detecting phishing attacks with several deep learning and optimization methods [51].

The suggested approach achieves an accuracy of 99.37%, a precision of 99.11%, and a sensitivity of 99.21% in the detection of phishing URLs. The experimental results demonstrate that the accuracy of the suggested method is higher than methods such as ODAE-WPDC, DL-SGD, DL-RMSProp, DL-Adam, SI-BBA, PDGAN, NIOSELM, MLP-SL, SVM-SL. Due to the balancing of the data set, the proposed method has a high ability to detect and classify phishing samples. The suggested approach is faster than the ODAE-WPDC method because, unlike ODAE-WPDC, it does not use meta-heuristic algorithms to optimize the parameters. The proposed method uses meta-heuristic methods to reduce the dimensions and the input of the CNN-LSTM neural network, increasing the learning speed in the proposed method.

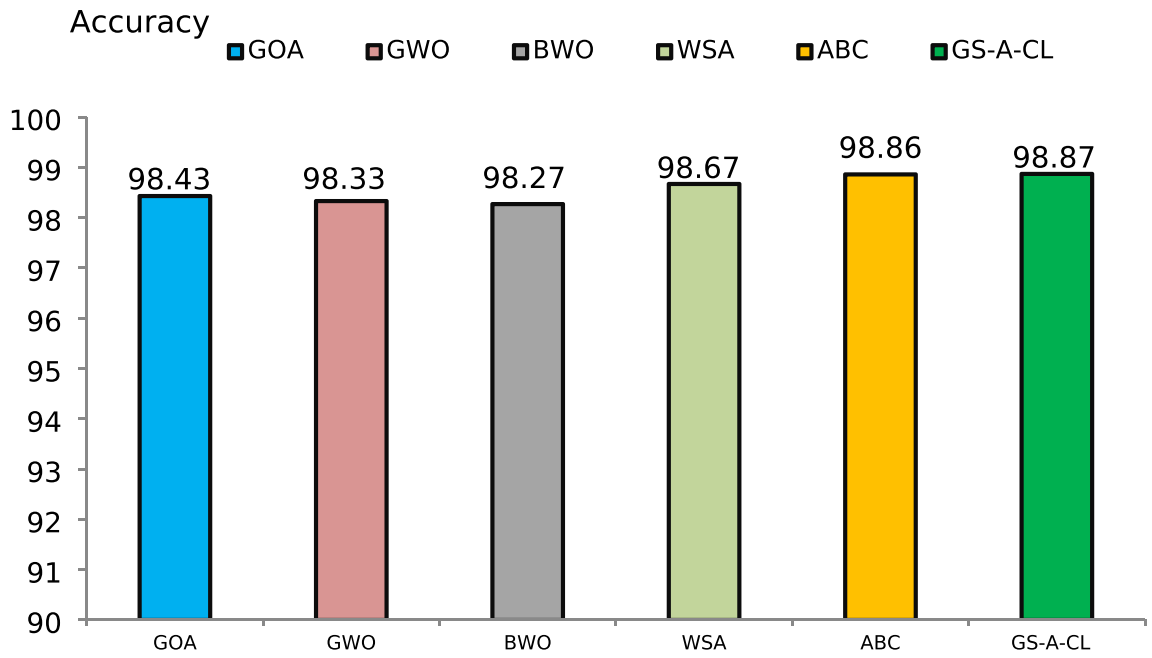

Fig. 21 Comparison of the accuracy of the proposed method in the UCI dataset with feature selection methods

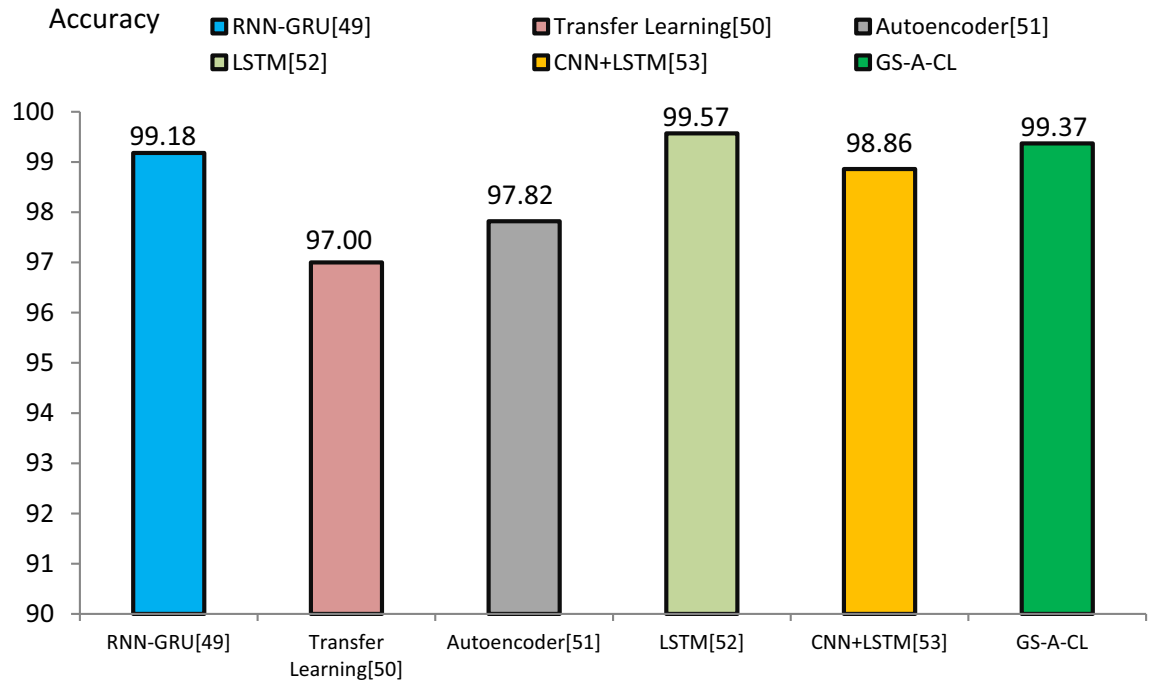

Fig. 22 Comparing the accuracy of the proposed method in the PhishTank dataset with deep learning methods

Table 4 Comparison of the proposed method with deep learning and meta-heuristic methods

Methods	Accuracy	Precision	Recall
ODAE-WPDC	99.28	99.29	99.24
DL-SGD	94.64	94.97	95.17
DL-RMSProp	92.84	93.77	95.34
DL-Adam	94.69	94.87	95.93
SI-BBA	94.93	94.59	94.84
PDGAN	94.12	94.96	94.02
NIOSELM	93.40	94.65	94.66
MLP-SL	87.80	88.75	87.41
SVM-SL	83.37	87.22	88.54
GS-A-CL	99.37	99.11	99.21

5 Conclusion

Phishing is a cyber-attack against Internet users with deception and social engineering. In phishing attacks, fake links are sent to users through email or other communication media and directed to fake websites. Phishing attacks are very harmful, and estimates show that these attacks cause millions of dollars in user losses. A practical approach is to use machine learning and deep learning methods for the detection of phishing attacks. CNN neural network is an efficient deep learning tool for pattern analysis and recognition. CNN neural network is used in most cases for image processing, and in a few types of research, this deep learning method has been used to classify text and URL strings. In the proposed method to detect phishing attacks, samples are first balanced. For sample balancing, the probabilistic GAN version is combined with the SMOTE method to perform deep learning on the balanced data set and reduce the learning error. The N-Gram method is used to extract the features, and the basic features are selected by the improved AVOA with chaos theory.

The chosen features are applied to the dataset, and the selected samples are transformed into input images for CNNs. In the final phase, an optimal and combined approach of CNN and LSTM is presented in detecting phishing pages and links. Experiments have been run on Phishtank, UCI, and Tan datasets. Experiments showed that the accuracy of the proposed method on UCI, PhishTank, and Tan datasets is 98.87%, 99.37%, and 98.79%, respectively. Experiments show that the Logistic chaotic function performs better to enhance the search discovery of the AVOA. The proposed approach demonstrates higher accuracy than the WSA, BWO, and ABC algorithms during the feature selection phase. Additionally, the suggested method outperforms the CNN, RNN-GRU, and Autoencoder methods in classifying samples.

The advantage of the proposed method is to balance the data set with two combined methods, and a more balanced data set is provided. Application of game theory in deep

learning based on conditional probabilities and more accuracy of this method than GAN is another advantage of the suggested approach. Another advantage of the suggested approach is to improve the African vulture algorithm with chaos theory in the feature selection phase. The suggested approach's advantage is its superior performance compared to similar meta-heuristic algorithms during the feature selection phase. An additional benefit of the suggested approach is the integration of CNN and LSTM architectures, which enhances the accuracy of classification in detecting phishing attacks.

The challenge of the proposed method is the high complexity and time overhead in the balancing and learning phase by the CNN-LSTM method and the need for optimization of LSTM and CNN parameters. The parameters of the CNN-LSTM will be optimized with meta-heuristic algorithms in future work. Another future work is providing an extension for browsers like Google Chrome to detect phishing attacks.

Author contributions M.A.E posed the idea and wrote part of the paper Ü.T.: analyzed and wrote part of the paper J.R.: implemented and simulated with matlab code J.M.L-G.: analyzed the results and modified the paper

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data availability Research Data Policy and Data Availability Statements: Restrictions apply to the availability of these data.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Greco, F., Desolda, G., Esposito, A.: Explaining phishing attacks: an XAI approach to enhance user awareness and trust. In: Proc. of the Italian Conference on CyberSecurity (ITASEC '23) (2023)
2. Buono, P., Desolda, G., Greco, F., Piccinno, A.: Let warnings interrupt the interaction and explain: designing and evaluating phishing email warnings. In: Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems, pp 1–6 (2023)
3. Marin, I.A., Burda, P., Zannone, N., Allodi, L.: The influence of human factors on the intention to report phishing emails. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp 1–18 (2023)
4. Xu, T., Singh, K., Rajivan, P.: Personalized persuasion: quantifying susceptibility to information exploitation in spear-phishing attacks. Appl. Ergon. **108**, 103908 (2023)
5. Lee, Y.Y., Gan, C.L., Liew, T.W.: Susceptibility to instant messaging phishing attacks: does systematic information processing differ between genders? Crime Prev. Community Saf. **25**(2), 179–203 (2023)
6. Syafitri, W., Shukur, Z., Asma'Mokhtar, U., Sulaiman, R., Ibrahim, M.A.: Social engineering attacks prevention: a systematic literature review. IEEE Access **10**, 39325–39343 (2022)
7. Kim, J., Lee, S., Kim, Y., Ahn, S., Cho, S.: Graph learning-based blockchain phishing account detection with a heterogeneous transaction graph. Sensors **23**(1), 463 (2023)
8. Atlam, H.F., Oluwatimilehin, O.: Business email compromise phishing detection based on machine learning: a systematic literature review. Electronics **12**(1), 42 (2022)
9. Gupta, B.B., Tewari, A., Jain, A.K., Agrawal, D.P.: Fighting against phishing attacks: state of the art and future challenges. Neural Comput. Appl. **28**, 3629–3654 (2017)
10. Jain, A.K., Gupta, B.B.: A novel approach to protect against phishing attacks at client side using auto-updated white-list. EURASIP J. Inf. Secur. **2016**, 1–11 (2016)
11. Gupta, A., Choudhary, G., Shandilya, S.K., Sihag, V.: A contemporary anti-phishing framework based on visual cryptography and steganography. Int. J. Internet Technol. Secur. Trans. **13**(2), 139–158 (2023)
12. 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)
13. Divakaran, D.M., Oest, A.: Phishing detection leveraging machine learning and deep learning: a review. arXiv Prepr. arXiv2205.07411 (2022)
14. Abdulrahman, L.M., Ahmed, S.H., Rashid, Z.N., Jghef, Y.S., Ghazi, T.M., Jader, U.H.: Web phishing detection using web crawling, cloud infrastructure and deep learning framework. J. Appl. Sci. Technol. Trends **4**(01), 54–71 (2023)
15. Lin, S.-C., Wl, P.-C., Chen, H.-Y., Morikawa, T., Takahashi, T., Lin, T.-N.: Senseinput: an image-based sensitive input detection scheme for phishing website detection. In: ICC 2022-IEEE International Conference on Communications, pp 4180–4186 (2022)
16. Feng, S., Keung, J., Zhang, P., Xiao, Y., Zhang, M.: The impact of the distance metric and measure on SMOTE-based techniques in software defect prediction. Inf. Softw. Technol. **142**, 106742 (2022)
17. Abdollahzadeh, B., Gharehchopogh, F.S., Mirjalili, S.: African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput. Ind. Eng. **158**, 107408 (2021)
18. Abdulghani Ali Ahmed, A.A.A., et al.: A honeybee-inspired framework for a smart city free of internet scams. Sensors **23**(4284), 1–14 (2023)
19. Kalabarige, L.R., Rao, R.S., Abraham, A., Gabralla, L.A.: Multilayer stacked ensemble learning model to detect phishing websites. IEEE Access **10**, 79543–79552 (2022)
20. Kaushik, K., Singh, S., Garg, S., Singhal, S., Pandey, S.: Exploring the mechanisms of phishing. Comput. Fraud Secur. **2021**(11), 14–19 (2021)
21. Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J.-N., Bayne, E., Bellekens, X.: Utilising deep learning techniques for effective zero-day attack detection. Electronics **9**(10), 1684 (2020)
22. Soltani, M., Ousat, B., Siavoshani, M.J., Jahangir, A.H.: An adaptable deep learning-based Intrusion detection system to zero-day attacks. J. Inf. Secur. Appl. **76**, 103516 (2023)

23. Guo, Y.: A review of machine learning-based zero-day attack detection: challenges and future directions. Comput. Commun. **198**, 175–185 (2023)
24. He, S., et al.: Combining deep learning with traditional features for classification and segmentation of pathological images of breast cancer. In: 2018 11th International Symposium on Computational Intelligence and Design (ISCID), vol. 1, pp 3–6 (2018)
25. Alabandi, G.A.: Combining Deep Learning with Traditional Machine Learning to Improve Classification Accuracy on Small Datasets (2017)
26. Xie, J., Jiang, H., Song, W., Yang, J.: A novel quality control method of time-series ocean wave observation data combining deep-learning prediction and statistical analysis. J. Sea Res. **195**, 102439 (2023)
27. Rajeswary, C., Thirumaran, M.: A comprehensive survey of automated website phishing detection techniques: a perspective of artificial intelligence and human behaviors. In: 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), pp 420–427 (2023)
28. Pandey, M.K., Singh, M.K., Pal, S., Tiwari, B.B.: Prediction of phishing websites using machine learning. Spat. Inf. Res. **31**(2), 157–166 (2023)
29. Alsenani, T.R., Ayon, S.I., Yousuf, S.M., Anik, F.B.K., Chowdhury, M.E.S.: Intelligent feature selection model based on particle swarm optimization to detect phishing websites. Multimed. Tools Appl. 1–33 (2023)
30. Sun, Y., Chong, N., Ochiai, H.: Federated phish bowl: LSTM-based decentralized phishing email detection. In: 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp 20–25 (2022)
31. Jha, B., Atre, M., Rao, A.: Detecting cloud-based phishing attacks by combining deep learning models. In: 2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA), pp 130–139 (2022)
32. Almousa, M., Zhang, T., Sarrafzadeh, A., Anwar, M.: Phishing website detection: How effective are deep learning-based models and hyperparameter optimization? Secur. Priv. **5**(6), e256 (2022)
33. Sharma, S.R., Singh, B., Kaur, M.: Improving the classification of phishing websites using a hybrid algorithm. Comput. Intell. **38**(2), 667–689 (2022)
34. Gupta, S., Bansal, H.: Trust evaluation of health websites by eliminating phishing websites and using similarity techniques. Concurr. Comput. Pract. Exp. **35**, e7695 (2023)
35. Yoo, J., Cho, Y.: ICSA: Intelligent chatbot security assistant using Text-CNN and multi-phase real-time defense against SNS phishing attacks. Expert Syst. Appl. **207**, 117893 (2022)
36. Trinh, N.B., Phan, T.D., Pham, V.-H.: Leveraging deep learning image classifiers for visual similarity-based phishing website detection. In: Proceedings of the 11th International Symposium on Information and Communication Technology, pp 134–141 (2022)
37. Liu, D.-J., Geng, G.-G., Zhang, X.-C.: Multi-scale semantic deep fusion models for phishing website detection. Expert Syst. Appl. **209**, 118305 (2022)
38. Remmide, M.A., Boumahdi, F., Boustia, N., Feknous, C.L., Della, R.: Detection of phishing URLs using temporal convolutional network. Procedia Comput. Sci. **212**, 74–82 (2022)
39. Shaik, C.M., Penumaka, N.M., Abbireddy, S.K., Kumar, V., Aravinth, S.S.: Bi-LSTM and conventional classifiers for email spam filtering. In: 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), pp 1350–1355 (2023)
40. Pham, T.D., Pham, T.T.T., Hoang, S.T., Ta, V.C.: Exploring efficiency of GAN-based generated URLs for phishing URL detection. In: 2021 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), pp 1–6 (2021)
41. Bu, S.-J., Cho, S.-B.: Integrating deep learning with first-order logic programmed constraints for zero-day phishing attack detection. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 2685–2689 (2021)
42. Mohanty, S., Acharya, A.A.: MFBFST: building a stable ensemble learning model using multivariate filter-based feature selection technique for detection of suspicious URL. Procedia Comput. Sci. **218**, 1668–1681 (2023)
43. Farida, F., Mustopa, A.: Comparison of logistic regression and random forest using correlation-based feature selection for phishing website detection. Sist. J. Sist. Inf. **12**(1), 13–20 (2023)
44. Thirumaran, M., Karthikeyan, R.P., Rathaamani, V.: Phishing website detection using natural language processing and deep learning algorithm. Adv. Sci. Technol. **124**, 712–718 (2023)
45. Sharma, A., Singh, P.K., Chandra, R.: SMOTified-GAN for class imbalanced pattern classification problems. Ieee Access **10**, 30655–30665 (2022)
46. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv Prepr. arXiv1412.6980 (2014)
47. Sun, Y., et al.: Energy theft detection model based on VAE-GAN for imbalanced dataset. Energies **16**(3), 1109 (2023)
48. 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)
49. Burgess, J., O'Kane, P., Sezer, S., Carlin, D.: LSTM RNN: detecting exploit kits using redirection chain sequences. Cybersecurity **4**(1), 1–15 (2021)
50. Tanha, J., Zarei, Z.: The Bombus-terrestris bee optimization algorithm for feature selection. Appl. Intell. **53**(1), 470–490 (2023)
51. Adebowale, M.A., Lwin, K.T., Hossain, M.A.: Intelligent phishing detection scheme using deep learning algorithms. J. Enterp. Inf. Manag. **36**(3), 747–766 (2023)
52. Somesha, M., Pais, A.R., Rao, R.S., Rathour, V.S.: Efficient deep learning techniques for the detection of phishing websites. Sādhanā **45**, 1–18 (2020)