Computers, Materials & Continua
DOI: 10.32604/cmc.2022.031625
Article

Tech Science Press

Hunger Search Optimization with Hybrid Deep Learning Enabled Phishing Detection and Classification Model

Hadil Shaiba[1], Jaber S. Alzahrani[2], Majdy M. Eltahir[3], Radwa Marzouk[4], Heba Mohsen[5] and Manar Ahmed Hamza[6,*]

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[2]Department of Industrial Engineering, College of Engineering at Alqunfudah, Umm Al-Qura University, Saudi Arabia
[3]Department of Information Systems, College of Science & Art at Mahayil, King Khalid University, Saudi Arabia
[4]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[5]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
[6]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
*Corresponding Author: Manar Ahmed Hamza. Email: ma.hamza@psau.edu.sa

Received: 22 April 2022; Accepted: 07 June 2022

Abstract: Phishing is one of the simplest ways in cybercrime to hack the reliable data of users such as passwords, account identifiers, bank details, etc. In general, these kinds of cyberattacks are made at users through phone calls, emails, or instant messages. The anti-phishing techniques, currently under use, are mainly based on source code features that need to scrape the webpage content. In third party services, these techniques check the classification procedure of phishing Uniform Resource Locators (URLs). Even though Machine Learning (ML) techniques have been lately utilized in the identification of phishing, they still need to undergo feature engineering since the techniques are not well-versed in identifying phishing offenses. The tremendous growth and evolution of Deep Learning (DL) techniques paved the way for increasing the accuracy of classification process. In this background, the current research article presents a Hunger Search Optimization with Hybrid Deep Learning enabled Phishing Detection and Classification (HSOHDL-PDC) model. The presented HSOHDL-PDC model focuses on effective recognition and classification of phishing based on website URLs. In addition, SOHDL-PDC model uses character-level embedding instead of word-level embedding since the URLs generally utilize words with no importance. Moreover, a hybrid Convolutional Neural Network-Long Short Term Memory (HCNN-LSTM) technique is also applied for identification and classification of phishing. The hyperparameters involved in HCNN-LSTM model are optimized with the help of HSO algorithm which in turn produced improved outcomes. The performance of the proposed HSOHDL-PDC model was validated using different datasets and the outcomes confirmed the supremacy of the proposed model over other recent approaches.

Keywords: Uniform resource locators; phishing; cyberattacks; machine learning; deep learning; hyperparameter optimization

1 Introduction

The rapid developments in information communication technologies and worldwide networks induced a paradigm shift from traditional working space to cyberspace, in terms of day-to-day activities such as e-commerce, electronic banking, social networking, and many more [1]. Anonymous, open, and uncontrolled structure of the internet is a splendid medium to make cyberattacks. It translates into the fact that not only the networks are prone to attacks, but the individual and experienced users too face such issues [2]. Though the users are experienced and cautious about cyberattacks, it has become impossible to prevent them from falling into phishing scam to the fullest. In order to increase the success rate of phishing attacks, cyber-attackers consider the personality features of the users too, in particular terms, to deceive the experienced users [3].

The analogy of the phishing attacks can be extracted from 'fishing' the victims. In recent times, these kinds of attacks grab high attention from the researchers. Attackers or phishers consider opening a few deceptive websites as an attractive and promising method. In these methods, exact famous model and legal sites over internet are reciprocated [4]. Even though such webpages consist of similar Graphical User Interfaces (GUI), it tend to have distinct Uniform Resource Locators (URLs) than the original page. Predominantly, a prudent and well-experienced user can identify such malignant web pages by just watching the URLs [5]. However, users tend to miss or not examine the complete address of their web page properly, due to hurried life style. These malicious webpages remain active and is usually sent through social networking tools, other web pages, or just by email messages.

Several anti-phishing methods have evolved in recent years to reduce the effect of phishing sites. Such methods are classified under four categories such as hybrid, lists, information flow method, and heuristics [6]. Among these, lists-related method includes two orders of lists such as blacklist and whitelist. While the former lists the phishing URLs and the latter lists legitimate URLs [7]. Heuristics-based method derives the characteristics of a page's URL and its content. It identifies the phishing sites through complete analysis of such features [8,9]. Hybrid method blends both heuristic-based and lists-based methods. Information flow technique adds bogus credentials to the original credential back and forth to a phishing website randomly [10]. Even though the researchers have enhanced feasible approaches to block phishing sites, attackers also evolved in the meantime to bypass recent tools and are able to deceive the victims.

The researchers in the study conducted earlier [11] projected a Machine Learning (ML) approach-based anti-phishing technique named as PHISH-SAFE based on URL feature. In order to estimate the efficacy of the suggested model, the study considered 14 features from URLs to differentiate whether a web page is phishing or not. The projected technique was trained upon 33,000 phishing and legitimate URLs using Naïve Bayes (NB) and Support Vector Machine (SVM) classifiers. Wang et al. [12] developed a rapid phishing website recognition method named PDRCNN based on URL of the website. It retrieves the content from target website or uses third-party service as a prior approach.

Barraclough et al. [13,14] proposed modern techniques combining web content-based, heuristic-based, and blacklist-based methodologies along with ML algorithms using comprehensive features to assist in proper recognition of phishing attacks. In the study conducted earlier [15–17], a technique based on Non-Inverse matrix Online Sequence Extreme Learning Machine (NIOSELM) was proposed to detect phishing attacks. This method considers three kinds of features to systematically describe a webpage. With the NIOSELM approach, Sherman Morriso Woodbury formula was used to prevent the matrix inversion function and presented the concept of Online Sequence Extreme Learning Machine (OSELM) to update the training module.

Ramana et al. [18] presented a smart technology using an ensemble of feature selection techniques to detect the phishing sites and achieve considerable results. The study employed different ML methods to find out the optimal classification method and proposed an ensemble technique using Extreme Gradient Boosting (XGBoost), Random forest, (RF), and Decision tree (DT) algorithms. In literature [19], an ML-based phishing recognition technique was presented to protect the webpage and users from cyber-attacks. In order to optimize the outcomes in an effective manner, Term Frequency-Inverse Document Frequency (TF-IDF) value of the website was applied with the technique. ML methodologies namely, Stochastic Gradient Descent (SGD), Logistic Regression (LR), RF, SVM, and NB were employed to train and test the attained data.

In this background, the current research article presents a Hunger Search Optimization with Hybrid Deep Learning enabled Phishing Detection and Classification (HSOHDL-PDC) model. The presented HSOHDL-PDC model focuses on effective recognition and classification of phishing based on website URLs. In addition, SOHDL-PDC model uses character-level embedding instead of word-level embedding since URLs generally use words of no importance. Moreover, a hybrid Convolutional Neural Network-Long Short Term Memory (HCNN-LSTM) technique is applied for identification and classification of phishing. Furthermore, the hyperparameters of HCNN-LSTM model are optimized with the help of HSO algorithm which in turn results in improved outcomes. The performance of the proposed HSOHDL-PDC model was validated using different datasets.

2 The Proposed Model

In this study, a novel HSOHDL-PDC model has been proposed for effective recognition and classification of phishing based on website URLs. The proposed HSOHDL-PDC model mainly utilizes character-level embedding rather than word-level embedding since URLs generally utilize words of no importance. Followed by, HSO is applied with HCNN-LSTM model for identification and classification of phishing. Fig. 1 illustrates the overall process of the proposed HSOHDL-PDC technique.

2.1 Character Level Embedding Process

In this work, HSOHDL-PDC model mainly utilizes character-level embedding instead of word-level embedding since URLs generally utilize words of no importance. URLs are processed at Character Level (CL) which remains a solution for difficult vocabulary. Data has been found to be comprised at CL level. The attacker simulates the URLs of original website by altering many unnoticeable characters. For sample, google.com can be altered to google.com by replacing ‘oo’ with ‘00’. CL embedding is utilized here to determine this derivative data which in turn enhances the efficiency of malicious URLs identification process. URLs are embedded by defining the m-sized alphabet to input language. Then, all the characters are embedded using one-hot encode. Next, the order of characters is changed to sequence these m-sized vectors at a fixed length, L.

- Tokenizer: Here, the tokenizer is utilized to proceed the URL from 'char level' and a token is added to the vocabulary. Afterwards, appropriate data is trained while the tokenizer comprises of all the essential info of the data.
- The vocabulary: The alphabets, utilized in general, contain 95 characters such as 10 numbers, 26 upper-case English letters, 26 lower-case English letters, and 33 other characters (e.g.,;.!?: ' /_@#$ . . . etc.)
- Character to index: Next the right vocabulary is received while every URL is demonstrated with the help of character index
- Padding: URL has to be at a distinct length and NN handles only the fixed-length vector. Thus, every URL is supposed to be equivalent length so that CNN procedure can be applied upon the batch data.

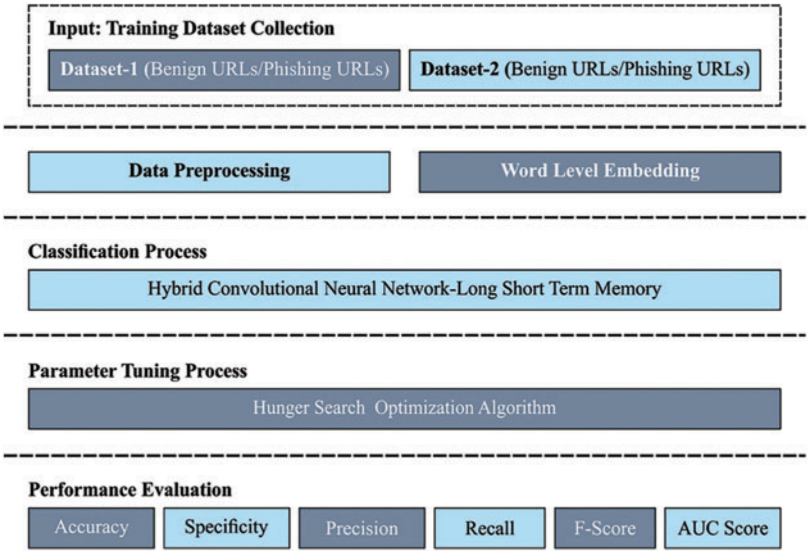
Input: Training Dataset Collection
Dataset-1 (Benign URLs/Phishing URLs)
Dataset-2 (Benign URLs/Phishing URLs)
Data Preprocessing
Word Level Embedding
Classification Process
Hybrid Convolutional Neural Network-Long Short Term Memory
Parameter Tuning Process
Hunger Search Optimization Algorithm
Performance Evaluation
Accuracy
Specificity
Precision
Recall
F-Score
AUC Score

Figure 1: Overall process of HSOHDL-PDC technique

2.2 Hybrid CNN-LSTM Based Classification Model

During data classification process, HCNN-LSTM model is utilized to recognize proper classes. Usually, Recurrent Neural Network (RNN) examines the input data for hidden consecutive designs. This is performed by concatenating the preceding data with present data in both spatial and temporal dimensions and forecasting the future sequence. While RNN extracts the hidden time-series pattern from consecutive data (for instance, video, sensor, or audio data), it is ineffective to remember or hold long data for a long period of time. Eventually, it fails in dealing the issues which involve long-term sequences. A similar kind of problem is signified by gradient exploding or vanishing gradient which is overcome using different types of RNNs, for instance Long Short Term Memory (LSTM). It has the ability to remember the data for a long period [20]. The internal structure of LSTM comprises of many gates (like output, input, and forget gates), whereas, during every iteration, the input in the preceding gate is forwarded to the next gate so as to control the flow of data near the last output. Every gate is generally measured by sigmoid or tanh activation function, i.e., the input gate i_t which is responsible for updating the data. But the value of h_t is calculated by scalar product of 0_t and tanh of C_t. Conversely, recurrent unit C_t evaluates the state of previous cell C_{t-1} and present input value x_t with the help of tanh activation functions. Fig. 2 depicts the framework of CNN-LSTM method.

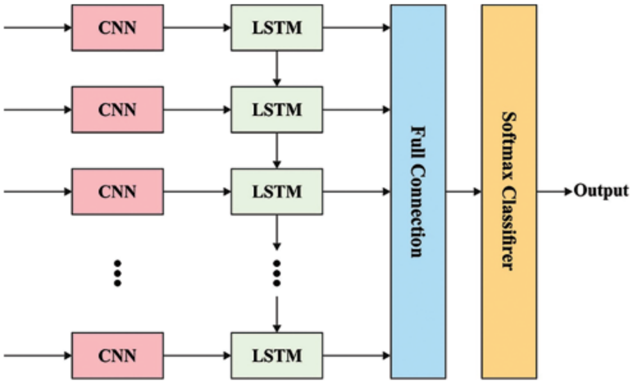

Figure 2: Structure of CNN-LSTM

At last, the ultimate result is achieved by passing h_t to softmax classification. Mathematically, the functions of the aforementioned gates are formulated herewith.

$$f_t = \Phi\left(\hat{W}_f \cdot [h_{t-1'}x_t] + B_f\right) \tag{1}$$

$$i_t = \Phi\left(\hat{W}_i \cdot [h_{t-1'}x_t] + B_i\right) \tag{2}$$

$$C_t = tanh\left(\hat{W}_C \cdot [h_{t-1}, x_t] + B_C\right) \tag{3}$$

$$C_t = f_t x C_{t-1} + i_t x C_t \tag{4}$$

$$0_t = \Phi\left(W_0 \cdot [h_{t-1'}x_t] + B_0\right) \tag{5}$$

$$h_t = o_t xtanh\left(\Phi\left(C_t\right)\right) \tag{6}$$

$$Output = softmax\left(h_t\right) \tag{7}$$

It can be presented in the name of HCNN-LSTM method, whereas the features are extracted in the layer of primary method and then forwarded to other methods for learning and modelling purposes. 1-CNN has been developed by the researchers to achieve excellent performance in terms of removing spatial and discriminative features from the data. But, LSTM is utilized by several researchers since it demonstrated its efficacy from sequential and time-series data. By searching these two models, the features with 1D-CNN can be extracted and then these features are forwarded for LSTM to learn and model. These features, found in CNN procedure, are then passed onto two LSTM layers of similar cell size i.e., 64 from all the layers. Adam optimization is utilized in this method with a learning rate of 0.0001.

2.3 Hyperparameter Optimization

Finally, the hyperparameter of HCNN-LSTM model are optimized with the help of HSO algorithm which in turn results in improved outcomes [21–24]. HSO algorithm is stimulated based on foraging and hunger behaviour of animals [25]. In case of (t) individual, its place is decided based on foraging performance and is demonstrated as a mathematical model below.

$$\overrightarrow{X(t+1)} = \begin{cases} Game_1: \overrightarrow{X(t)} \times (1 + randn(1)). & r_1 < l_{HGS} \\ Game_2: \overrightarrow{W_1} \times \overrightarrow{X_b} + \overrightarrow{R} \times \overrightarrow{W_2} \times |\overrightarrow{X_b} - \overrightarrow{X(t)}|. & r_1 > l_{HGS}.r_2 > E \\ Game_3: \overrightarrow{W_1} \times \overrightarrow{X_b} - \overrightarrow{R} \times \overrightarrow{W_2} \times |\overrightarrow{X_b} - \overrightarrow{X(t)}| & r_1 > l_{HGS}.r_2 < E \end{cases} \tag{8}$$

The factor $\overrightarrow{X(t)} \times (1 + randn(1))$ processes the individual searching for food, at the existing place itself, with an arbitrary hunger performance. In the meantime, the factor $|\overrightarrow{X_b} - \overrightarrow{X(t)}|$ demonstrates the t^{th} individual's activity range and the multiplication by factor $\overrightarrow{W_2}$ put on the effect of hunger on individual's activity. In order to control the activity of the individual, the term $\overrightarrow{R}$ is established. If R is slow and equal to 0, then it denotes that the individual is no longer hungry and its activities are halted. Then, the term $\overrightarrow{W_1} \times \overrightarrow{X_b}$ is added or subtracted to simulate the individual that their peer has reached the place of food. This simulation motivates the individual to search for food in their existing place. Here, the term $\overrightarrow{W_1}$ implies the error experienced by the individual in obtaining the actual place of the food. In order to calculate the difference in terms of controlling from every position, the subsequent formula is utilized.

$$E = sech\,(|F\,(i) - BF) \tag{9}$$

whereas $i = 1, 2, \cdots, n$. Also, sech $(x) = \frac{2}{e^x + e^{-x}}$. $\overrightarrow{R}$ is computed utilizing the subsequent equation,.

$$\overrightarrow{R} = 2 \times shrink \times rand - shrink \tag{10}$$

$$shrink = 2 \times \left(1 - \frac{t}{T}\right) \tag{11}$$

In problem space, HGS functions on the basis of logic of search as given herewith.

- Searching based on $\overrightarrow{X}$: The 1st game processes the individual's independent effort to search for the food, out of hunger. This is an approach that is non-cooperative with another individual.
- Searching based on $\overrightarrow{X_b}$: Both 2nd and 3rd games process the cooperation amongst the individuals by means of shared data, assuming the place of food. By tuning the variables $\overrightarrow{R}$, $\overrightarrow{W_1}$, and $\overrightarrow{W_2}$, the places of the individuals are upgraded based on the fundamental determination of other individuals.

At this point, the individual hunger $\overrightarrow{W_1}$, in Eq. (8) is modeled utilizing the following method.

$$\overrightarrow{W_1} = \begin{cases} hungry(i) \times \frac{N}{SHungry} \times r_4. & r_3 < l_{HGS} \\ 1. & r_3 > l_{HGS} \end{cases} \tag{12}$$

In the meantime, the other hunger $\overrightarrow{W_2}$ as in Eq. (11) is denoted by the equation given below.

$$\overrightarrow{W_2} = \left(1 - e^{(-|hungey(i) - SHungey|)}\right) \times r_5 \times 2 \tag{13}$$

Here, 'hungry' signifies the *hunger* of all the individuals.

For calculating the term $hungry(i)$ was utilized:

$$hungry\,(i) = \begin{cases} 0. & AllFitness\,(i) == BF \\ hungry\,(i) + H. & AllPitness\,(i)\,! == BP \end{cases} \tag{14}$$

whereas $AllFitness(i)$ signifies the fitness value of every individual from this iteration. In order to process some more iterations, the hunger value of the optimum individual is fixed at 0. The equation of (H) is projected in Eqs. (15) and (16).

$$TH = \frac{F(i) - BF}{WP - BF} \times r_6 \times 2 \times (UB - LB) \tag{15}$$

$$H = \begin{cases} LH \times (1 + r). & TH < LH \\ TH. & TH \geq LH \end{cases} \tag{16}$$

The factor $F(i) - BP$ implies the count of food required by the i^{th} individual to satisfy their hunger. It can be value variations with all the iterations. In the meantime, the factor $WP - BP$ defines the capacity of the individual to search for food. Hunger ratio is calculated through $\frac{F(i)-\mathrm{BF}}{WF-\mathrm{BF}}$. At last, the factor $r_6 \times 2$ establishes the positive or negative effect of factors from the neighboring environment on the individual's hunger.

HSO system develops a Fitness Function (FF) to obtain enhanced classification performance. Optimum solution translates into lesser error rate whereas the worst solution means enhanced error rate. FF is provided herewith.

$$fitness\,(x_i) = ClassifierErrorRate\,(x_i) = \frac{number\ of\ misclassified\ samples}{Total\ number\ of\ samples} * 100 \tag{17}$$

3 Experimental Validation

The proposed HSOHDL-PDC model was experimentally validated using two datasets [26]. The details of the dataset are provided in Tab. 1.

Table 1: Dataset description

Dataset	Benign URLs	Phishing URLs	Total No. of URLs
Dataset-1	36400	37175	73575
Dataset-2	43189	40668	83857

Fig. 3 illustrates a set of confusion matrices generated by the proposed HSOHDL-PDC model on dataset-1. With entire dataset, the proposed HSOHDL-PDC model categorized 35,820 samples under benign class and 36,935 samples under phishing class. Along with that, with 70% of training (TR) dataset, HSOHDL-PDC approach categorized 25,008 samples under benign class and 25,898 samples under phishing class. Further, with 30% of testing (TS) dataset, the presented HSOHDL-PDC technique categorized 10,817 samples under benign class and 11,037 samples under phishing class.

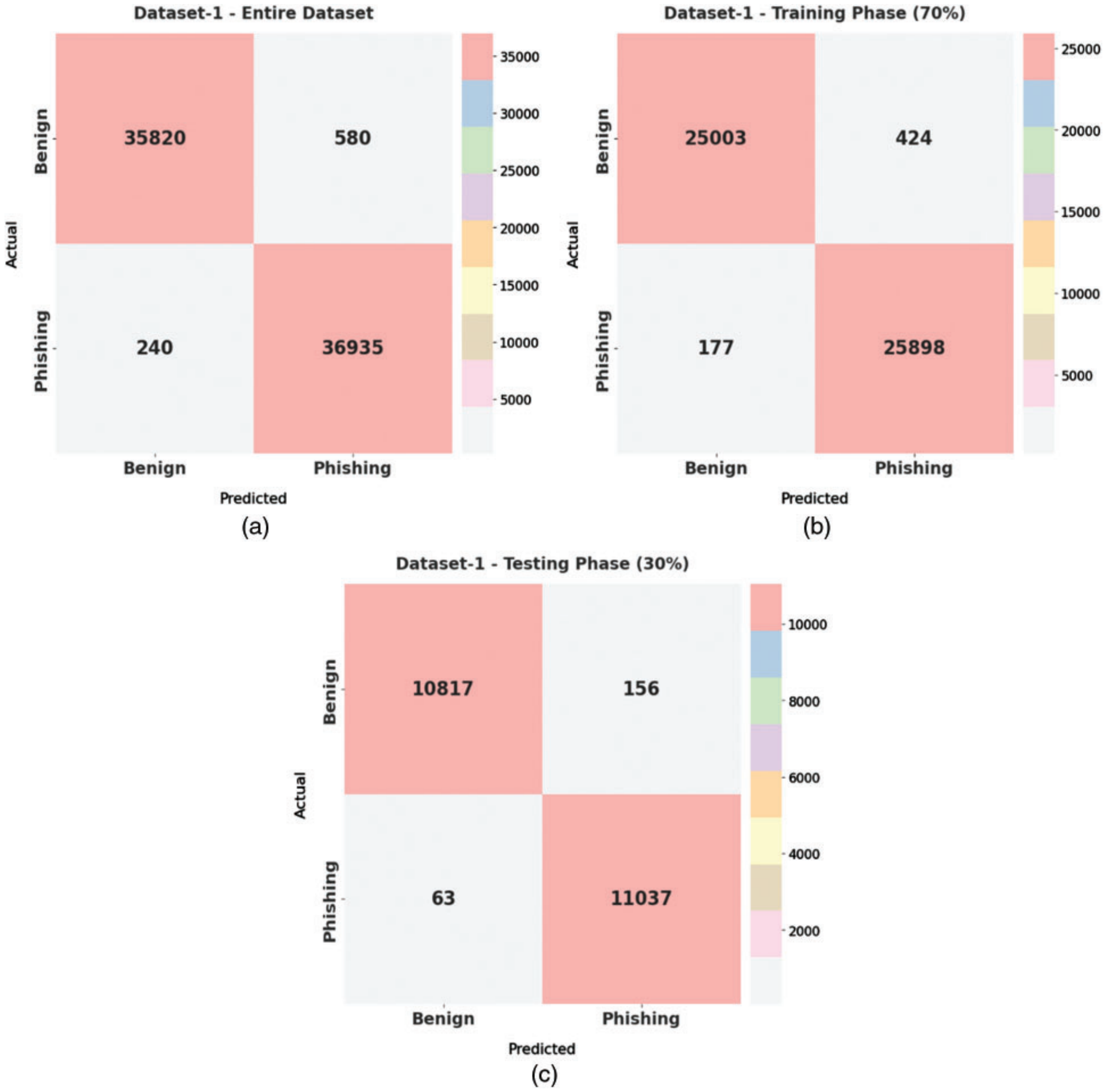

Figure 3: Confusion matrices generated by HSOHDL-PDC technique on dataset-1 (a) Entire dataset, (b) 70% of TR, and (c) 30% of TS

Tab. 2 and Fig. 4 report the overall classification analysis results achieved by the proposed HSOHDL-PDC model on test dataset-1. The table values imply that the proposed HSOHDL-PDC model produced improved outcomes under all the cases. For instance, with entire dataset, HSOHDL-PDC model provided average $accu_y$, $prec_n$, $reca_l$, $spec_y$, $F1_{score}$, and Area Under the Curve (AUC) values such as 98.89%, 98.89%, 98.88%, 98.88%, 98.89%, and 98.88% respectively. Moreover, with 70% of TR dataset, HSOHDL-PDC approach obtained average $accu_y$, $prec_n$, $reca_l$, $spec_y$, $F1_{score}$, and AUC values such as 98.83%, 98.84%, 98.83%, 98.83%, 98.83%, and 98.83% correspondingly. Furthermore, with

30% of TS dataset, HSOHDL-PDC technique offered average $accu_y$, $prec_n$, $reca_l$, $spec_y$, $F1_{score}$, and AUC values such as 99.01%, 99.01%, 99.01%, 99.01%, 99.01%, and 99.01% correspondingly.

Table 2: Results of the analysis of HSOHDL-PDC technique under different measures on dataset-1

Dataset-1						
Classes	Accuracy	Precision	Recall	Specificity	F-score	AUC score
Entire dataset						
Benign	98.89	99.33	98.41	99.35	98.87	98.88
Phishing	98.89	98.45	99.35	98.41	98.90	98.88
Average	98.89	98.89	98.88	98.88	98.89	98.88
Training phase (70%)						
Benign	98.83	99.30	98.33	99.32	98.81	98.83
Phishing	98.83	98.39	99.32	98.33	98.85	98.83
Average	98.83	98.84	98.83	98.83	98.83	98.83
Testing phase (30%)						
Benign	99.01	99.42	98.58	99.43	99.00	99.01
Phishing	99.01	98.61	99.43	98.58	99.02	99.01
Average	99.01	99.01	99.01	99.01	99.01	99.01

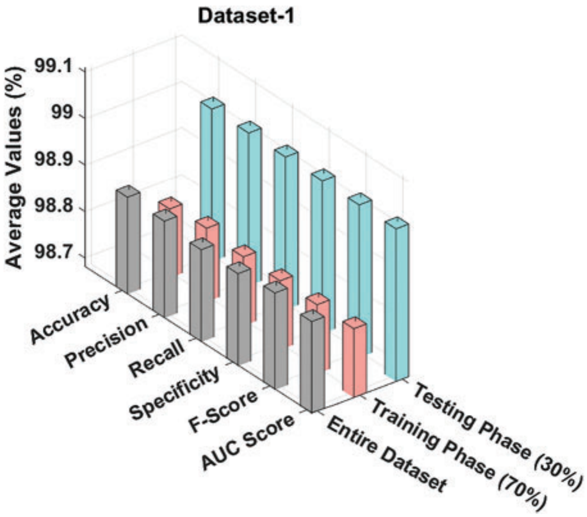

Figure 4: Results of the analysis of HSOHDL-PDC technique under different measures on dataset-1

A brief precision-recall examination was conducted upon HSOHDL-PDC technique on test dataset-1 and the results are portrayed in Fig. 5. By observing the figure, it can be inferred that the proposed HSOHDL-PDC model accomplished the maximum precision-recall performance under all the classes.

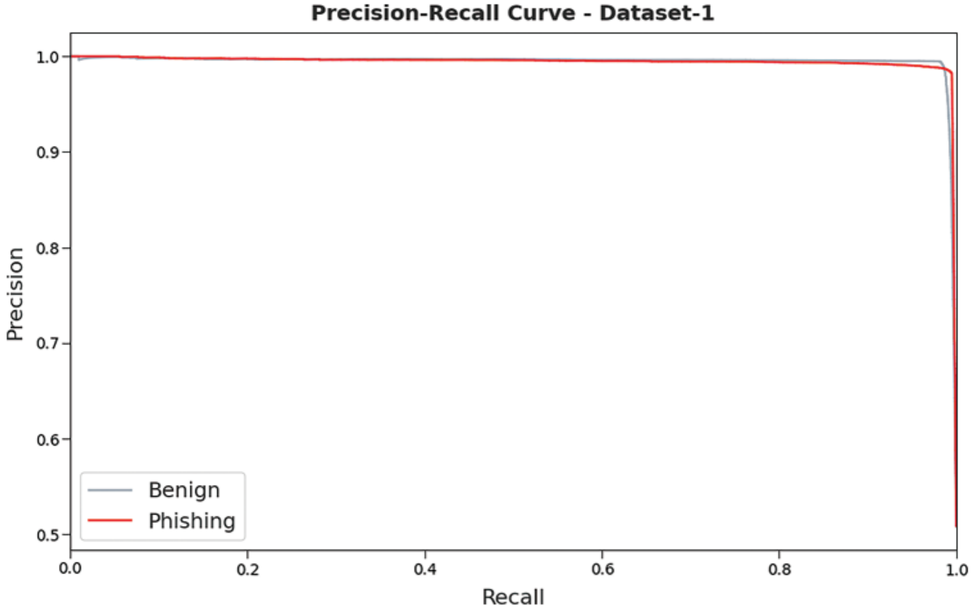

Figure 5: Precision-recall curve analysis results of HSOHDL-PDC technique on dataset-1

A detailed ROC investigation was conducted upon HSOHDL-PDC method on test dataset-1 and the results are portrayed in Fig. 6. The results imply that the proposed HSOHDL-PDC model exhibited its ability to differentiate two distinct classes such as 'benign' and 'phishing' on test dataset.

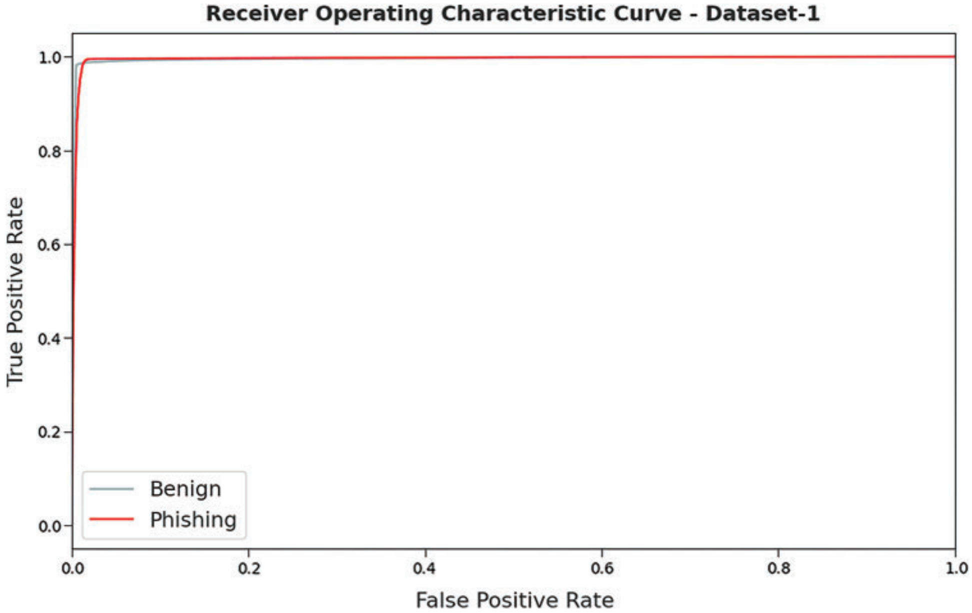

Figure 6: ROC curve analysis results of HSOHDL-PDC technique on dataset-1

Figs. 7 and 8 provide an overview on comprehensive comparative analysis results accomplished by HSOHDL-PDC and other recent models [27]. The results indicate that Gaussian NB model achieved the least performance over other methods. Followed by, multinomial NB, LR, RF, and XGBoost models showcased moderately closer classification performance. Moreover, Deep Neural Network (DNN) and CNN models demonstrated reasonable outcomes with $accu_y$ values being 95.24% and 95.41% respectively. However, the proposed HSOHDL-PDC model accomplished maximum $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and AUC values such as 99.01%, 99.01%, 99.01%, 99.01%, and 99.01% respectively.

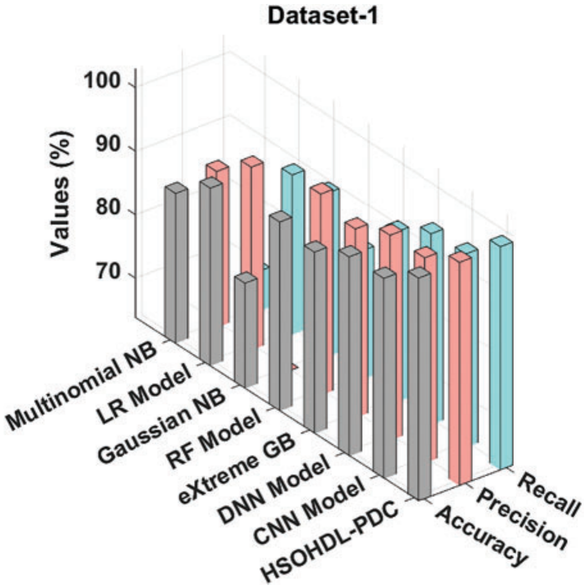

Figure 7: Comparative analysis results of HSOHDL-PDC technique on dataset-1

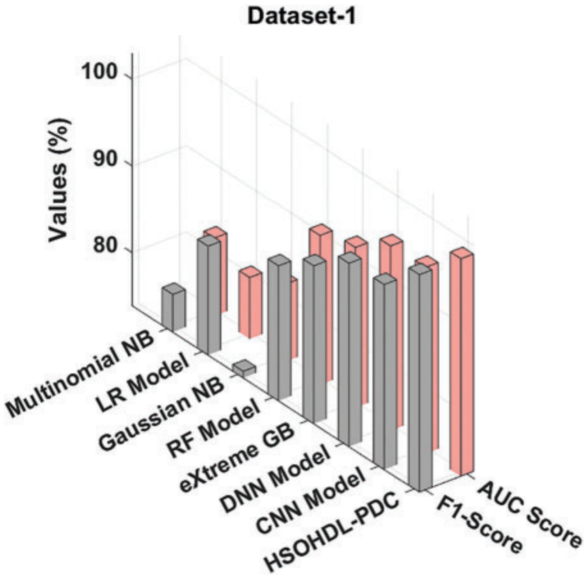

Figure 8: AUC and F1-score analyses results of HSOHDL-PDC technique on dataset-1

Fig. 9 shows a set of confusion matrices generated by HSOHDL-PDC technique on dataset-2. With entire dataset, the proposed HSOHDL-PDC model categorized 42,348 samples under benign class and 39,973 samples under phishing class. In addition, with 70% of TR dataset, the proposed HSOHDL-PDC model categorized 29,675 samples under benign class and 27,973 samples under phishing class. Eventually, with 30% of TS dataset, HSOHDL-PDC algorithm categorized 12,673 samples under benign class and 12,000 samples under phishing class.

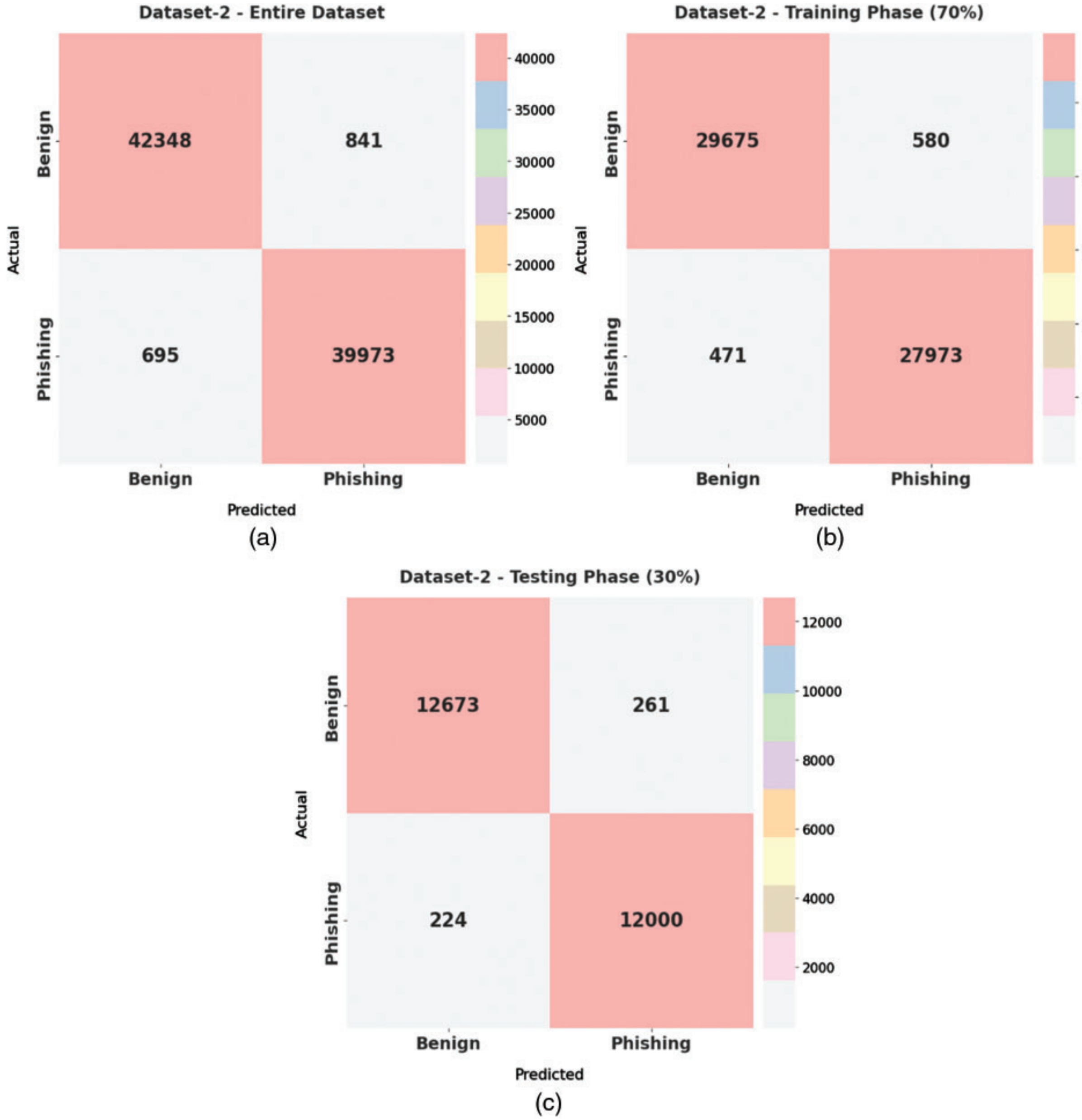

Figure 9: Confusion matrices generated by HSOHDL-PDC technique on Dataset-2 (a) Entire dataset, (b) 70% of TR, and (c) 30% of TS

Tab. 3 and Fig. 10 demonstrate the overall classification output accomplished by HSOHDL-PDC system on test dataset-2. The table values infer that the proposed HSOHDL-PDC model resulted in improved outcomes under all the cases. For instance, with entire dataset, HSOHDL-PDC model offered average $accu_y$, $prec_n$, $reca_l$, $spec_y$, $F1_{score}$, and AUC values such as 98.17%, 98.16%, 98.17%, 98.17%, 98.17%, and 98.17% respectively. Followed by, with 70% of TR dataset, HSOHDL-PDC method attained average $accu_y$, $prec_n$, $reca_l$, $spec_y$, $F1_{score}$, and AUC values such as 98.21%, 98.20%, 98.21%, 98.21%, 98.21%, and 98.21% correspondingly. Moreover, with 30% of TS dataset, the proposed HSOHDL-PDC technique provided average $accu_y$, $prec_n$, $reca_l$, $spec_y$, $F1_{score}$, and AUC values such as 98.07%, 98.07%, 98.07%, 98.07%, 98.07%, and 98.07% correspondingly.

Table 3: Results of the analysis of HSOHDL-PDC technique under different measures on Dataset-2

Dataset-2						
Classes	Accuracy	Precision	Recall	Specificity	F-score	AUC score
Entire dataset						
Benign	98.17	98.39	98.05	98.29	98.22	98.17
Phishing	98.17	97.94	98.29	98.05	98.11	98.17
Average	98.17	98.16	98.17	98.17	98.17	98.17
Training phase (70%)						
Benign	98.21	98.44	98.08	98.34	98.26	98.21
Phishing	98.21	97.97	98.34	98.08	98.16	98.21
Average	98.21	98.20	98.21	98.21	98.21	98.21
Testing phase (30%)						
Benign	98.07	98.26	97.98	98.17	98.12	98.07
Phishing	98.07	97.87	98.17	97.98	98.02	98.07
Average	98.07	98.07	98.07	98.07	98.07	98.07

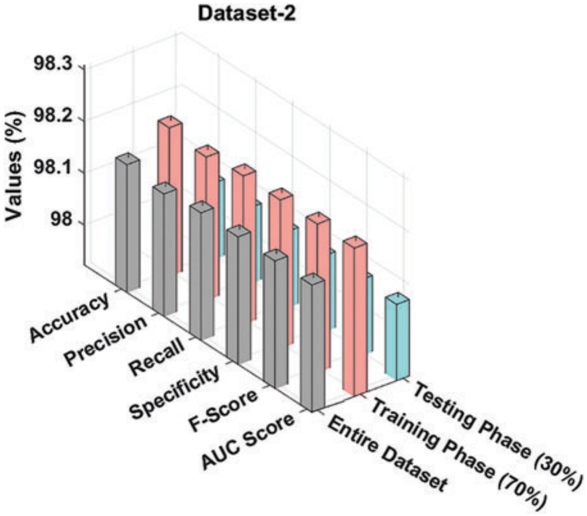

Figure 10: Results of the analysis of HSOHDL-PDC technique under different measures on Dataset-2

A detailed precision-recall examination was conducted upon HSOHDL-PDC model on test Dataset-2 and the results are shown in Fig. 11. By observing the figure, it can be understood that the proposed HSOHDL-PDC technique accomplished the maximum precision-recall performance under all the classes.

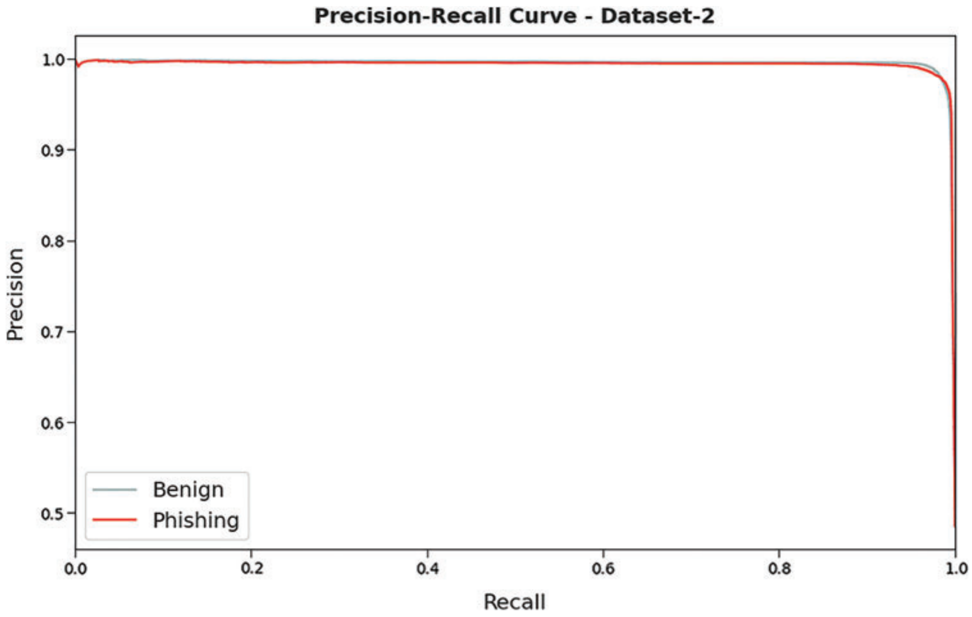

Figure 11: Precision-recall curve analysis results of HSOHDL-PDC technique on dataset-2

A brief Receiver Operating Characteristic (ROC) analysis was conducted upon HSOHDL-PDC method on test dataset-2 and the results are shown in Fig. 12. The outcomes imply that the proposed HSOHDL-PDC approach exhibited its ability to differentiate two different classes such as ‘benign’ and ‘phishing’ on the test dataset.

Tab. 4 shows the comprehensive comparative analysis results accomplished by the proposed HSOHDL-PDC technique and other recent techniques. The outcomes imply that Gaussian NB approach produced the least performance over other methodologies. Next, multinomial NB, LR, RF, and XGBoost models showcased moderately closer classification performance. Besides, DNN and CNN algorithms exhibited reasonable outcomes with $accu_y$ values being 95.23% and 95.34% respectively. At last, the proposed HSOHDL-PDC model accomplished reasonable outcomes with maximal $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and AUC values such as 98.07%, 98.07%, 98.07%, 98.07%, and 98.07% correspondingly.

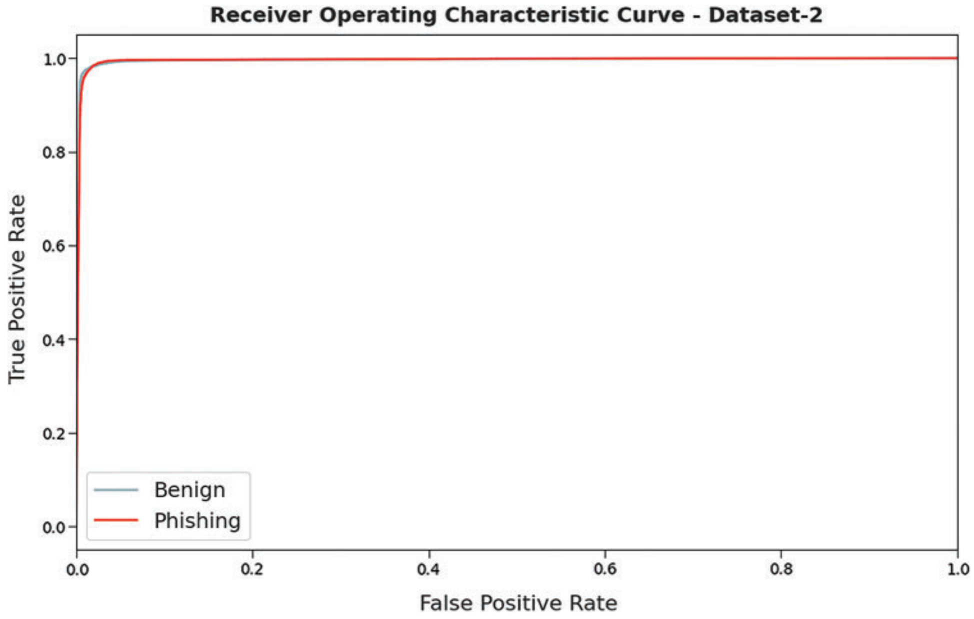

Figure 12: ROC curve analysis results of HSOHDL-PDC technique on dataset-2

Table 4: Comparative analysis results of HSOHDL-PDC technique and other existing approaches on Dataset-2

Methods	Accuracy	Precision	Recall	F1-score	AUC score
Multinomial NB	96.54	95.04	94.25	94.83	95.83
LR Model	94.93	96.23	94.88	95.37	94.20
Gaussian NB	95.29	94.11	96.96	95.77	96.30
RF Model	96.92	94.42	96.15	95.96	96.11
eXtreme GB	95.49	94.79	94.05	96.12	94.07
DNN Model	95.23	94.89	96.78	95.87	94.35
CNN Model	95.34	95.30	94.57	94.51	94.78
HSOHDL-PDC	98.07	98.07	98.07	98.07	98.07

4 Conclusion

In this study, a novel HSOHDL-PDC model has been developed for effective recognition and classification of phishing based on website URLs. The proposed HSOHDL-PDC model mainly utilizes character-level embedding instead of word-level embedding since URLs generally utilize words of no importance. Followed by, HSO is applied with HCNN-LSTM model for identification and classification of phishing. Furthermore, the hyperparameters of HCNN-LSTM model are optimized with the help of HSO algorithm which in turn results in improved outcomes. The performance of the proposed HSOHDL-PDC model was validated using different datasets and the outcomes confirmed the supremacy of the proposed model over recent approaches. In future, hybrid HSO algorithm can be applied to enhance the performance of he HSOHDL-PDC model.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (158/43). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R135), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 22UQU4340237DSR22.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] O. Sahingoz, E. Buber, O. Demir and B. Diri, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, no. 4, pp. 345–357, 2019.

[2] M. Zouina and B. Outtaj, "A novel lightweight URL phishing detection system using SVM and similarity index," *Human-Centric Computing and Information Sciences*, vol. 7, no. 1, pp. 17, 2017.

[3] G. Sonowal and K. S. Kuppusamy, "PhiDMA – A phishing detection model with multi-filter approach," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 1, pp. 99–112, 2020.

[4] H. Y. A. Abutair and A. Belghith, "Using case-based reasoning for phishing detection," *Procedia Computer Science*, vol. 109, no. 1, pp. 281–288, 2017.

[5] A. Butnaru, A. Mylonas and N. Pitropakis, "Towards lightweight url-based phishing detection," *Future Internet*, vol. 13, no. 6, pp. 154, 2021.

[6] B. Wei, R. A. Hamad, L. Yang, X. He, H. Wang *et al.,* "A deep-learning-driven light-weight phishing detection sensor," *Sensors*, vol. 19, no. 19, pp. 4258, 2019.

[7] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, no. 13, pp. 153–166, 2019.

[8] T. Chin, K. Xiong and C. Hu, "Phishlimiter: A phishing detection and mitigation approach using software-defined networking," *IEEE Access*, vol. 6, pp. 42516–42531, 2018.

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

[10] J. Mao, W. Tian, P. Li, T. Wei and Z. Liang, "Phishing-alarm: Robust and efficient phishing detection via page component similarity," *IEEE Access*, vol. 5, pp. 17020–17030, 2017.

[11] A. K. Jain and B. B. Gupta, "PHISH-SAFE: URL features-based phishing detection system using machine learning," in *Cyber Security*, Advances in Intelligent Systems and Computing book series, Springer, Singapore, vol. 729, pp. 467–474, 2018.

[12] W. Wang, F. Zhang, X. Luo and S. Zhang, "PDRCNN: Precise phishing detection with recurrent convolutional neural networks," *Security and Communication Networks*, vol. 2019, no. 6, pp. 1–15, 2019.

[13] P. A. Barraclough, G. Fehringer and J. Woodward, "Intelligent cyber-phishing detection for online," *Computers & Security*, vol. 104, no. 2, pp. 102123, 2021.

[14] M. A. Hamza, S. B. Haj Hassine, I. Abunadi, F. N. Al-Wesabi, H. Alsolai *et al.,* "Feature selection with optimal stacked sparse autoencoder for data mining," *Computers, Materials & Continua*, vol. 72, no. 2, pp. 2581–2596, 2022.

[15] L. Yang, J. Zhang, X. Wang, Z. Li, Z. Li *et al.,* "An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features," *Expert Systems with Applications*, vol. 165, no. 2, pp. 113863, 2021.

[16] A. A. Albraikan, S. B. Haj Hassine, S. M. Fati, F. N. Al-Wesabi, A. M. Hilal *et al.,* "Optimal deep learning-based cyberattack detection and classification technique on social networks," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 907–923, 2022.

[17] A. M. Hilal, J. S. Alzahrani, I. Abunadi, N. Nemri, F. N. Al-Wesabi *et al.,* "Intelligent deep learning model for privacy preserving IIoT on 6g environment," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 333–348, 2022.

[18] A. V. Ramana, K. L. Rao and R. S. Rao, "Stop-Phish: An intelligent phishing detection method using feature selection ensemble," *Social Network Analysis and Mining*, vol. 11, no. 1, pp. 110, 2021.

[19] E. S. Vishva and D. Aju, "Phisher fighter: website phishing detection system based on url and term frequency-inverse document frequency values," *Journal of Cyber Security and Mobility*, 2021, http://10.13052/jcsm2245-1439.1114.

[20] I. U. Khan, S. Afzal and J. W. Lee, "Human activity recognition via hybrid deep learning based model," *Sensors*, vol. 22, no. 1, pp. 323, 2022.

[21] A. Muthumari, J. Banumathi, S. Rajasekaran, P. Vijayakarthik, K. Shankar *et al.,* "High security for de-duplicated big data using optimal simon cipher," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 1863–1879, 2021.

[22] R. Gopi, P. Muthusamy, P. Suresh, C. G. G. S. Kumar, I. V. Pustokhina *et al.,* "Optimal confidential mechanisms in smart city healthcare," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 4883–4896, 2022.

[23] D. A. Pustokhin, I. V. Pustokhin, P. Rani, V. Kansal, M. Elhoseny *et al.,* "Optimal deep learning approaches and healthcare big data analytics for mobile networks toward 5G," *Computers & Electrical Engineering*, vol. 95, no. 11, pp. 107376, 2021.

[24] J. A. Alzubi, O. A. Alzubi, M. Beseiso, A. K. Budati and K. Shankar, "Optimal multiple key-based homomorphic encryption with deep neural networks to secure medical data transmission and diagnosis," *Expert Systems*, vol. 39, no. 4, pp. e12879, 2022.

[25] P. Mehta, B. S. Yildiz, S. M. Sait and A. R. Yildiz, "Hunger games search algorithm for global optimization of engineering design problems," *Materials Testing*, vol. 64, no. 4, pp. 524–532, 2022.

[26] R. Rao, T. Vaishnavi and A. Pais, "CatchPhish: Detection of phishing websites by inspecting URLs," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 813–825, 2019.

[27] A. Aljofey, Q. Jiang, Q. Qu, M. Huang and J. Niyigena, "An effective phishing detection model based on character level convolutional neural network from url," *Electronics*, vol. 9, no. 9, pp. 1514, 2020.