Phishing detection model using Feline Finch Optimization based LSTM classifier

Abstract: Phishing detection technologies are essential to ensuring that users have a secure online experience by preventing users from falling prey to online fraud, divulging personal information to an attacker, and other risks. In order to stop email scams, this research employs the FFO-based LSTM classifier to identify phishing websites. The FFO method is used to pick and optimize the relevant parameters from many parameters that the LSTM offers. This reduces the computational complexity of the system and improves performance. The LSTM classifier conducts a more thorough examination of the network, which aids in enhancing the effectiveness of phishing detection. The proposed FFO-based LSTM classifier achieved the exceptional values of 93.16%, 94.59%, and 92.66%, which illustrates the efficiency of the research. The accuracy, sensitivity, and specificity of the metrics are utilised to illustrate the significance of the research.

Keywords*:* Phishing, LSTM classifier, FFO algorithm, SMOTE, synthetic data.

1. Introduction

Communication of networks is highly enhanced by the Internet of Things (IoT) along with that device connectivity is also enhanced because communication plays a major role in recent advancements. Smart gadgets are interconnected to exchange information, carry out computations, and keep track of various real-time scenarios [1]. Security continues to be one of the fundamental issues for communication and engagement, just as technology is developing [2][1]. Cybercrime incidences are rising quickly in direct proportion to the exponential expansion of Internet users [3]. The increased occurrence of cyber crimes, and cyber attacks greatly affected the end users by initiating various vulnerabilities and threats in the cyber-space. The effects of unauthorized users compromising data in cyberspace include information loss and financial damage [4] [5]. Every day, both individuals and organizations lose millions of dollars due to the occurrence of cyber threats [6] [3]. Cybersecurity events have been prevalent in recent years due to the fact that the attackers have successfully breached government systems including those of the US State Department and the White House [6], corporations including Google and RSA [7], websites of politicians and social organizations from various nations including John Podesta and the Democratic National Committee [8] in the majority of these incidents [9]. The increasing power and popularity of phishing attempts are shown by this string of high-profile events. On the other hand, phishing emails frequently result in financial harm to businesses [9].

Phishing is a popular cyber-attack technique that involves creating fraudulent websites that appear to be trustworthy in an effort to fraudulently collect sensitive data [10]. Phishing is a type of cybercrime, where malicious links are sent through spam or social media platforms under the false illusion of legitimate ones to trick users into visiting and giving up their private information. The attackers can then use this information to steal money and also for various other illegal activities. One of the major hazards to people's daily networking situations right now is phishing. The Anti-Phishing Working Group (APWG) reported that around 180768 phishing attacks were found worldwide in the first quarter of 2019 [11] [12]. Phishing can be performed in different ways, which are enlisted as follows: 1) randomly disseminating spam with phishing links to the individuals 2) stealing personal properties by impersonating legitimate websites, like those for online transactions 3) creating fraudulent e-commerce websites with false commodity information to steal money from purchases; and 4) sending deceptive messages with phishing links to users by some social software [13] [10]. Despite the fact that various anti-phishing solutions have been used, phishing attempts seem to be highly increased [10] [14].

To combat phishing attempts, numerous solutions have been invented and put forth by researchers and cybersecurity specialists [15][16]. The blacklist-based phishing attack detection approach is one of the important solutions established for tackling this issue. The blacklist-based method verifies the legitimacy of the requested universal resource locator (URL) by comparing it to URLs that have been gathered from phishing websites. This method categorically relies on a list of phishing URLs that have been flagged as malicious by cyber security professionals [17] [12]. On the other hand, machine learning (ML)-based approaches were developed to counteract the phishing attacker's dynamic modifications by detecting the legitimacy of a website using features taken from such websites [5] [18]. To represent phishing URLs and connected websites, common characteristics are initially gathered, including details about URLs, website architecture, and JavaScript features. Then, using the chosen features, phishing datasets are gathered. The underlying classifiers are subsequently trained to recognize the phishing websites. The Bayes model, Support Vector Machine (SVM), Association Roles, Logistic Regression (LR), neural networks, and others are examples of classifiers that are used for phishing detection [2]. The machine learning methods can automatically learn to classify data accurately, but it may also include a significant amount of duplicate points in publicly available phishing datasets with negative and worthless attributes, trapping the machine learning approach in the over-fitting issue [11].

The main aim of the research is to identify the phishing attack by predicting the normal and abnormal messages using FFO based LSTM classifier. The necessity for the development of a phishing detection model is focused on in this paper and the steps involved in this phishing detection model are preprocessing, data stability, and prediction using LSTM. The significant contribution of the research is interpreted as follows,

- ***Feline Finch Optimization algorithm:*** The FFO algorithm is developed by the standard hybridization of the Cat Swarm Optimization (CSO) [19] and Sparrow Search Optimization (SSO) [20], where the tracking and the foraging characteristics of finches and felines are combined for the effective optimizing of LSTM classifier.
- ***FFO based LSTM classifier:*** The LSTM classifier is used for the prediction of phishing attacks that takes place through offensive mails, where the long-term dependence characteristics help in the effective identification of the normal and abnormal messages. A large number of learning parameters present in the LSTM classifier helps in reducing the complexity and the application of the FFO algorithm effectively tunes the parameters for boosting the convergence of the classifier.
- ***SMOTE:*** The problems aroused due to the data imbalance are effectively resolved using SMOTE by providing higher data stability.

The flow of the manuscript is interpreted as follows: Section 2 shows the detailed analysis of the methods involved in phishing attack detection by enumerating the advantages, disadvantages, and challenges. Section 3 provides the methodology used for the detection of phishing websites along with the FFO based LSTM model. Section 4 reveals the results obtained using the proposed phishing detection model. Finally, the results are concluded in section 5.

2. Motivation

A phishing attack is a crucial factor that plays a major role in threatening the individual’s information, which results in heavy financial losses and various other factors. The various types of research based on the phishing detection performed using different techniques are interpreted in the below section for gaining sufficient knowledge.

2.1 Literature review

The methods established by the previous researchers relevant to the phishing website detection are enumerated in this section: Erzhou Zhu *et al.* [11] developed a decision tree and used optimal features for resolving the issues aroused due to the overfitting issues present in the neural network classifier. After resolving the overfitting issue, the neural networks are utilised for the identification of phishing websites but there is a lack of method needed for performing the optimal feature selection.

Gunikhan Sonowala and K S Kuppusamy [3] developed a multi-layer model for the detection and categorization of different phishing attacks. The accessible interface in the model helped the visually impaired people to access this model without any difficulties but only a few parameters are used in the model, which degraded the performance.

Shan Wang *et al.* [1] used two classifiers such as BiLSTM and random forest for the phishing detection that acted as a secure medium for the communication channel and enhanced the security by avoiding phishing websites. The disadvantage of the research is that the classifier was affected by the overfitting issues.

Qi Li *et al.*[9] dwelled with big email data and detected phishing messages using the LSTM classifier. The sizes of the data are improved by a k-means algorithm that satisfied the necessity of data in the deep learning technique. The lack of an accurate labeled set is also overcome by using the feature extraction algorithm but there is an occurrence of loss in the number of features.

Liqun Yang *et al.*[10] used extreme learning machine for the phishing detection and used variant techniques for the reduction of dependency of the model reduced the imbalanced distribution of the samples and the dimension of the data are reduced for reducing the complexity. Although the method achieved good performance the accuracy obtained is not sufficient and there is an increased time complexity.

L. Lakshmi *et al.* [21] identified phishing using hyperlinks and differentiated the fraudulent websites using a neural network model. The method provided high accuracy in phishing detection but was not suitable for large datasets.

Ammar Odeh *et al.*[22] used highly correlated features for the identification of phishing websites and was performed using multiple classifiers based on the adaptive boosting approach. The time taken for the selection of the correlated features and building of the model is comparably low but the presence of noise degrades the performance of the classifier.

Victor E. Adeyemo *et al.*[5] introduced an ensemble-based logistic model for the detection of phishing websites that worked more efficiently. The method was more resistant and flexible to varying biases but not preferred for high-dimensional data.

2.2 Challenges

The challenges to be overcame while detecting the phishing website is discussed below.

- The emerging type of phishing attack on the internet cannot be detected by phishing detection systems relying on user education [22]. This initiates a need for a more efficient detection model for phishing websites.

- The most prevalent phishing detection techniques based on visual resemblance take more time and initiate more difficulties. Additionally, the method could involve a lot of computing loads leading to complexity in phishing detection.
- To achieve considerable performance, several phishing detection systems depending on deep learning techniques need a comprehensive feature selection and classification algorithm. The scale of the final classifiers will also increase if the features are excessive, but the detection accuracy of the final classifiers will significantly decrease if the features are negative or meaningless [11][22]. Hence the classifier should overcome the above-mentioned difficulties for better performance.
- Some phishing attacks contain a significant number of duplicate variables in public phishing datasets, together with irrelevant and undesirable characteristics in the feature vectors, capturing the machine learning approach in the over-fitting or data imbalance issues [11].
- Due to issues with feature dimension, the machine learning modes are incredibly inefficient when dealing with a massive amount of data and result in the loss of a significant number of features. Second, because phishing emails are so flexible, attackers can easily alter their header features and bypass machine learning-based phishing email detection systems [9].

3. Research methodology for the detection of phishing websites using the optimized LSTM classifier:

The ultimate intention of this research is to detect the phishing websites based on deep learning techniques, namely FFO based LSTM classifier is used for the detection of the phishing website by detecting the offensive emails. At first, the input data from the phishing database [23] [24] is collected and the data is preprocessed for the removal of noisy and irrelevant data. The SMOTE is applied to the preprocessed data for reducing imbalanced issues and performing data stabilization. The stabilized data is fed forward to the FFO based LSTM classifier, where the prediction of the phishing websites is performed by the identification of offensive emails. The significant contribution of the research relies on the FFO algorithm developed by the standard hybridization of the characteristics of the feline and finches that optimize the classifier by tuning the parameters such as weights and bias of the LSTM classifier, which helps in achieving better convergence while predicting the phishing websites. The implementation is carried out using the software Python and the metrics, such as accuracy, False Acceptance Ratio (FAR), and False Rejection Ratio (FRR) are analyzed to reveal the efficacy of the proposed method. The schematic representation of the methodology is depicted in figure 1.

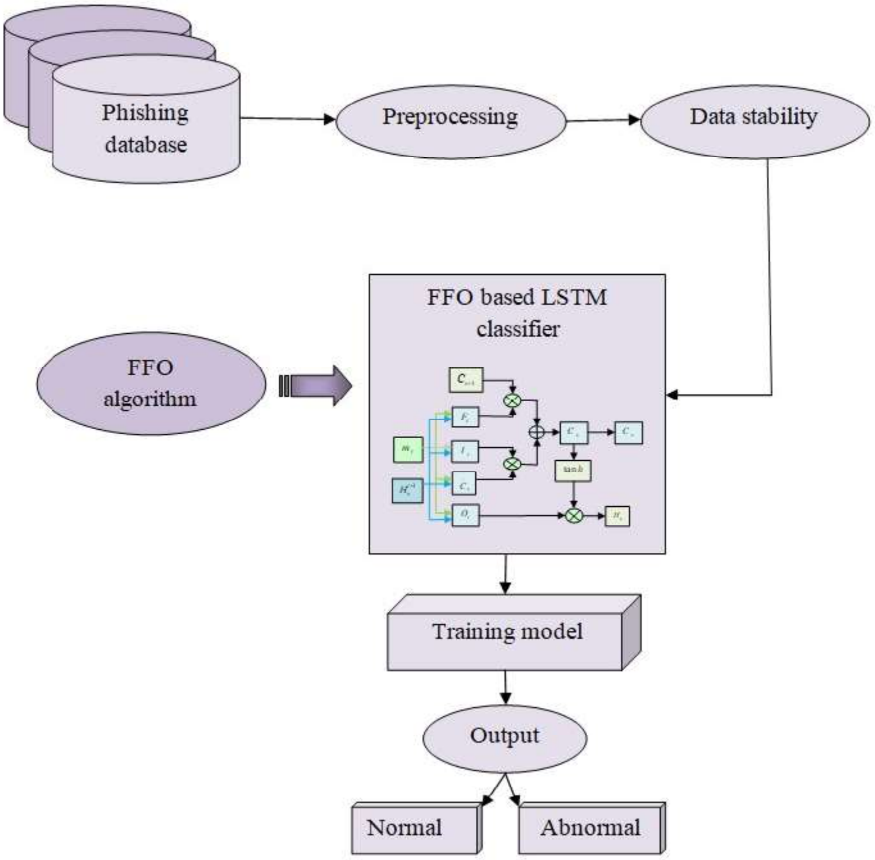


Figure 1. Representation of proposed FFO based LSTM phishing detection model

3.1 Input data

The input for the detection of malicious emails is gathered from the phishing database and is mathematically represented as follows,

$$P = \sum_{i=1}^{x} P_i \tag{1}$$

here, P denotes the phishing database and the number of attributes preset in the database is given by i in the range $1,2,3......x$.

3.2 Preprocessing of data

In order to detect phishing, the raw data must be transformed into organized data through the process of data preprocessing and is mathematically expressed as follows,

$$X = \{X_1, X_2, X_3, ...X_i,\} \tag{2}$$

Here, X denotes the preprocessed data.

3.3 Data imbalance problem using SMOTE:

The data imbalance issues have the capability to degrade the performance of the classifier and to avoid this the SMOTE is applied for the purpose of stabilization of data, which is described briefly in the below section.

3.3.1 SMOTE: SMOTE is the process of generating synthetic data points for resolving the issues arouses due to instability. The generation of synthetic data points helps in the oversampling of the minority class and the steps involved in initiating data stability using SMOTE are enumerated as follows: Relying upon the necessity of the oversampling, the nearest neighbors are chosen in a random manner. As an initial step, the difference between the nearest neighbors and the sample considered is determined. The determined difference is multiplied by a random number generated between 0 and 1. The resultant multiplication is added to the original vector so that the minority class could be balanced. The data couldn't be separated into testing and training after applying SMOTE, and if this happens the resultant output leads to misclassifications and there is a possibility of the reputation of the data. This could be avoided by applying the SMOTE after splitting the data into testing and training, which overcomes the imbalance issues.

3.4 FFO based LSTM classifier for the detection of phishing website

The LSTM classifier effectively predicts the phishing website by using the long-term dependence characteristics; along with that the vanishing gradient problem is also effectively resolved. The occurrence of the vanishing gradient initiates difficulties in the testing and training of data by the classifier, which is eliminated using the LSTM classifier. The hyper parameters such as weights and bias of the classifier are furthermore optimized using the FFO algorithm that also helps in tuning the classifier to identify the phishing websites. The LSTM consists of input layer that consists of the input from the phishing database [23] [24] and these data are converted into a fixed-length vector by the embedding layers. The input is then subjected to the LSTM units and then fed forwarded to the hidden layers. The evaluation of the LSTM is performed based on the cell state C_s, where the irrelevant information is forgotten by the forgetting gate and the essential information alone is stored in the memory for the evaluation of phishing web pages and this information will be the output of the hidden state layers H_s. The cell state is nothing but the long-term memory used by the LSTM classifier. The memory, forgetting and the output are controlled by the forgetting gate and the output gate and memory gate are controlled by the hidden layers. The block diagram representation of the LSTM classifier is depicted in figure 2.

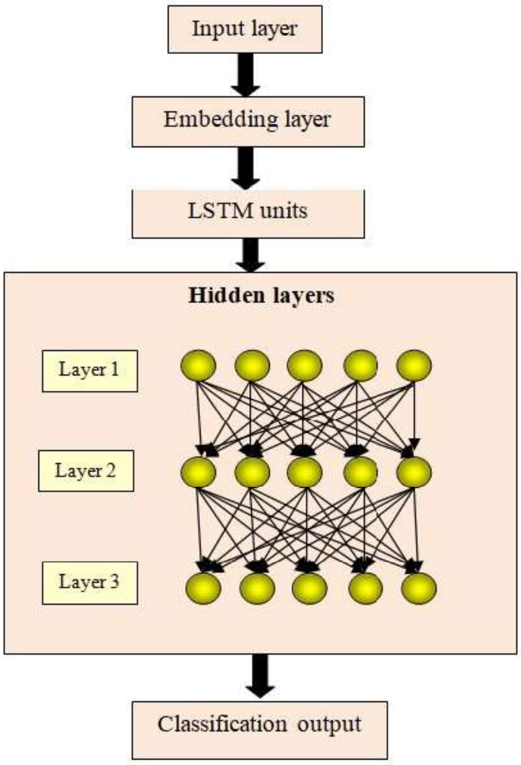


Figure 2. Architecture of LSTM classifier

Each LSTM unit consists of the input gate, the forgetting gate, and the output gate represented by I_t, F_t and O_t. The input data given to the LSTM unit is represented by I_d, the cell state is denoted by C_s and the temporary cell state is denoted by $\tilde{C}_s$.

$$I_t = A\left(W_I\left[H_s^{t-1}, m_t\right] + B_I\right) \tag{3}$$

$$F_t = A\left(W_F\left[H_s^{t-1}, m_t\right] + B_F\right) \tag{4}$$

$$O_t = A\left(W_O\left[H_s^{t-1}, m_t\right] + B_O\right) \tag{5}$$

here, the attributes W and B denotes the weight and bias of the LSTM classifier and H_s^{t-1} stands as the output of the individual LSTM units. The input of the current LSTM unit is given by m_t . The block diagram representation of the individual LSTM unit is observed in figure 3.

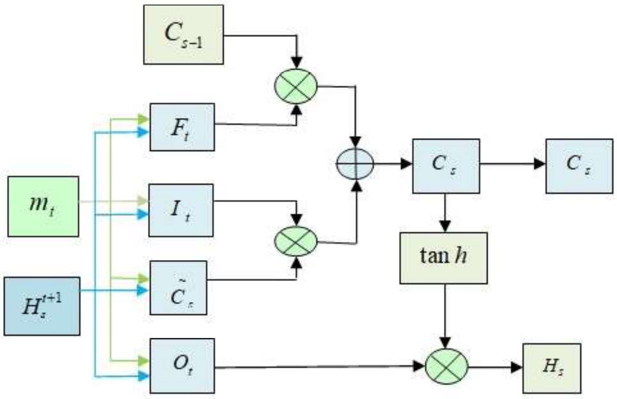


Figure 3. Architecture of single unit LSTM

3.4.1 Feline finch optimization

The felines are identified as the solutions and the number of solutions to be identified is predetermined. Mimicking the characteristics of the feline the optimization of the classifier is carried out and these characteristics are furthermore enhanced by the hybridization of the characteristics of finches, which boosts the performance by providing higher search space and faster convergence rate.

***Motivation**:* The felines (Cats) belong to the family Felis Catus and there are around 32 different species of felines. The anatomy of the feline is similar to that of other felid species, which has a robust, flexible body, rapid reflexes, pointed teeth, and retractable claws that are designed to destroy small prey. It has a keen sense of smell and good eye sight. Meowing, purring, trilling, hissing, growling, and grunting are just a few of the vocalizations and body language used by felines to communicate. The feline is a solitary hunter but a gregarious animal, a nocturnal predator that is most active at dawn and dark. It is capable of hearing sounds emitted by mice and other small mammals that are too faint or high in frequency for human ears. Despite the fact that all felines are quite curious, they are typically not very active. Felines have a very high level of alertness even while they are napping, they stay alert. These characteristics are used for the optimization of the classifier and the convergence of the classifier is enhanced by the foraging characteristics of the gregarious finches (Sparrow).

Mathematical model for the Feline Finch Optimization: In order to boost the speed of the phishing website detection and to stabilize the process of training the FFO is implemented and the steps involved in the FFO optimization are mathematically expressed in the below sections.

Initialization of solution: Each individual in the population of the feline is considered a candidate solution and the fitness value of each feline is evaluated to evaluate the global best solution and the candidate in the feline group is represented by,

$$F\left(c\right)=\sum f_1,f_2 \tag{6}$$

here, $F\left(c\right)$ denotes the number of candidate solutions initiated in the feline and finch group. The attribute f_1 represents the feline group and the attribute f_2 denotes the group of finches.

i) Look and rest mode

The felines keep on rest at the same time they keep an eye on the environment. Whenever the felines identify an attack or danger the felines choose their next course of iteration, and if they fixed to change their position, they move very slowly and cautiously. While sleeping the feline observes N directional solution space and they shift their position after awakening. Since the feline is aware of the surroundings the movement of the feline takes place based on the Feline seeking memory pool (FSMP), Range of selected dimension (RSD), Number of changes in dimension (NCD), and Ready to change (RTC). The look and rest mode could be used in the classifier for the recognition of the data as phishing or normal.

FSMP: The size of the dimension that is observed by each feline is described in FSMP through the points defined by the felines.

RSD: The mutative ratio for the chosen dimensions is provided by RSD. In the rest and seeking node if the dimension gets changed the variation between the new dimension and the old dimension will be in the range defined by RSD.

NCD: The number of changes in the dimensions carried by the felines is given by NCD.

RTC: The RTC is a Boolean variable that denotes the feline standing ready for the change in the dimension. The value of RTC does not have any impact on FSMP.

The steps involved in the look and rest node of the feline is enumerated as follows: Initially, the RTC is evaluated. If the value of RTC is true then the felines alter their positions based on their memory represented by $z = FSMP$. If the value of RTC is false then there will not occur any movements and is given by $z = FSMP - 1$. Secondly, the current position of the feline f_1^n is copied z times. Relying upon the NCD, the percent of the RSD gets increased or decreased.

Finally, the fitness value is evaluated for all the candidate solutions for felines. If the evaluated fitness values are not equal then the probability of the fitness values is selected based on the equation interpreted below,

$$\Phi(j) = \frac{F_j - F_q}{F_{\max} - F_{\min}} \qquad ; 0 < j < z \tag{6}$$

here, Φ denotes the probability of the fitness values, F represents the fitness function, F_q is the fitness function that can be varied by the user. If there is a need for a minimal solution then $F_q = F_{\min}$ and if there is a need for a maximum solution then $F_q = F_{\max}$ is used.

ii) Tracking mode

The tracking mode defines the characteristics of chasing the prey. When the target is spotted by the felines the speed and direction of the felines changes with respect to the target speed and direction. Hence, the velocity of the feline f_1 in the dimension d is given by,

$$S_{f_1,d} = S_{f_1,d} + rand \times q\left(P_{best,d} - P_{f_1,d}\right) \tag{7}$$

here, the best position of the feline is given by $P_{best,d}$ and the position of the feline is given by $P_{f_1,d}$. d is the dimension in the range $1,2,3,.......l$. The attribute c represents constant value and the random value assigned is represented by $rand$. $P_{f_1,d}$ is given by $P_{f_1,d} = P_{f_1,d} + S_{f_1,d}$. The tracking mode helps in providing faster convergence by detecting phishing attacks.

iii) Monitoring mode

The finches are intelligent birds classified into two groups such as dependent finches and independent finches. The dependent finches depend on the independent finches for their food, and continuously monitor the independent finches for obtaining the food. The finches that are continuously monitored will fight for their food by leaving their current position and getting the food before the other finches, which is considered as the best solution. The finches obtain the best solution by updating their position and their position update is mathematically expressed as,

$$P_{f_2,d} = \begin{cases} G \cdot \exp\left(\dfrac{P^i_{worst} - P^i_{f_2,d}}{f_2^2} \right) & ; if \left(f_2 > {}^{n}\!/\!_{2} \right) \\ P^{i+1}_{best} + | P^i_{f_2,d} - P^{i+1}_{best} | \cdot Q^+ \cdot H & ; else \end{cases} \tag{8}$$

here, G represents a random number that obeys normal distribution, Q denotes the matrix of dimension $1 \times d$, and $Q^+ = Q^T - \left(QQ^T \right)^{-1}$, where each element in the matrix will be 1. P^i_{worst} denotes the current worst solution obtained during the search by the independent finch and P_{best} represents the best optimal solution in the search space found by the independent finches, that is the position where the prey is determined. If the condition $f_2 > {}^{n}\!/\!_{2}$ is satisfied, then the f_2^{th} finch is considered to be starving more and i denotes the number of iterations.

iv) Enhancing mode

The tracking mode of felines is limited to a certain range, which is considerably low, and when the continuous monitoring and search made by the finches for their food is integrated with the tracking characteristics of felines the search space is expanded along with that the monitoring capacity of the felines are also improved while combining with finches. This expansion helps in the identification of attacks and enhances the performance of the classifier to obtain the global optimal solution. The integration is performed by concerning [25] and the enhanced performance is mathematically notified by the combination of (7) and (8) as follows,

$$\Phi_{best} = 0.5\,S_{f_1,d} + 0.5\,P_{f_2,d} \tag{9}$$

Resolving the above equation by the appliance of the respected values the resultant will be,

$$\Phi_{best} = 0.5\left\{\left[S_{f_1,d} + rand \times q\left(P_{best,d} - P_{f_1,d}\right)\right] + \left[P_{best}^{i+1} + |\, P_{f_2,d}^{i} - P_{best}^{i+1} \,| \cdot Q^{+} \cdot H\right]\right\} \tag{10}$$

here, $S_{f_1,d}$ denotes the speed of the feline to capture the prey in tracking mode and $P_{f,d}$ denotes the position update of the finches with respect to the position of food, which provides large search space so that faster convergence and global solution could be attained. The algorithmic procedure involved in the FFO algorithm is enumerated in table 1.

Table 1: Pseudocode for the proposed FFO algorithm

S.No	Pseudocode for the proposed FFO algorithm
1	Input: f_1, f_2
2	Output: Φ_{best}
3	Evaluate: ***FSMP, RSD, NCD, RTC*** **# Look and rest mode**
4	Evaluate memory
5	if ($RTC = 1$) then $z = FSMP$
6	if ($RTC = 0$) then $z = FSMP - 1$
7	Evaluate fitness: F
8	if ($F = f_1^n$); Assign F
9	if ($F \neq f_1^n$); Determine $\Phi\left(j\right)$

10	Determine velocity: $S_{f_1,d}$	**# Tracking mode**
11	Position update of finches: $P_{f,d}$	**# Monitoring mode**
12	Determine global best solution: Φ_{best}	**# Enhancing mode**
13	Terminate	

4. Results and discussion

The results attained using the FFO based LSTM classifier while detecting phishing websites are described in detail in the below sections.

4.1 Dataset description

The dataset utilized for phishing website detection is the Web page Phishing Detection Dataset [24] and Phishing Website Detection by Machine Learning Techniques dataset [23].

4.1.1 Web page Phishing Detection Dataset

A phishing website is a widely used social engineering technique that imitates reliable URLs and websites. The dataset consists of necessary URL- and website content-based attributes of both phishing and benign URLs of websites. The dataset consists of URL, length of the URL, length of the hostname, number of dots, and's, and so on. The features such as address bar-based features, domain based features, HTML & Javascript based features are extracted from this dataset.

4.1.2 Phishing Website Detection by Machine Learning Techniques dataset

The dataset comprised of 11430 URLs and around 87 features are extracted. The features come from three different classes: seven are extracted through contacting external services, while the remaining 56 are taken from the structure and syntax of URLs. The collection is evenly distributed; it contains exactly 50% legitimate URLs and 50% phishing URLs.

4.2 Parameter metrics

The parameter metrics used for proving the significance of the research are accuracy, sensitivity, and specificity and are detailed in the below sections.

Accuracy: The number of instances that are correctly identified as phishing websites by the proposed FFO based LSTM classifier is termed as accuracy and is given by,

$$Accu = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Sensitivity: The probability of detecting the phishing websites by the proposed FFO based LSTM classifier is measured using sensitivity and is given by,

$$Sens = \frac{TP}{TP + FN} \tag{12}$$

Specificity: The probability of detecting the nonphishing websites correctly is termed as specificity and is given by,

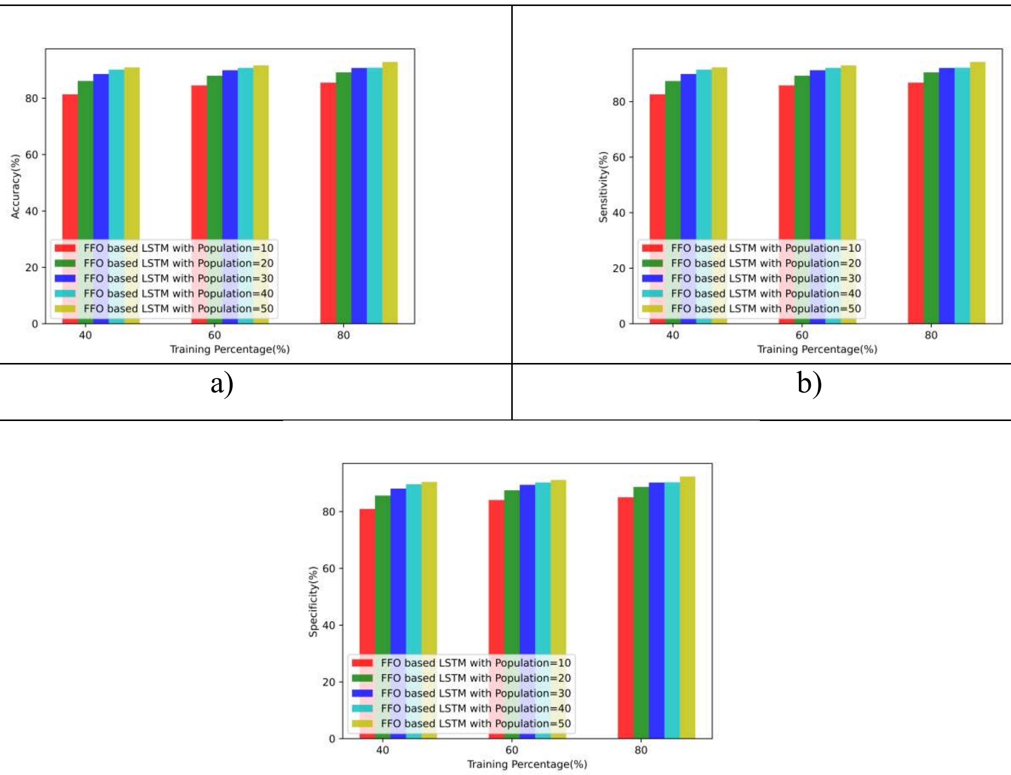
$$Spec = \frac{TN}{TN + FP} \tag{13}$$

4.3 Performance analysis of the proposed FFO based LSTM classifier

The performance analysis is performed based on the parameter metrics accuracy, sensitivity, and specificity for varying epochs and is interpreted in detail in the below sections.

4.3.1 Performance analysis based on dataset-1 concerning the training percentage

The performance analysis is made using dataset-1 and the observation is shown in figures 4 a), b), and c) in terms of accuracy, sensitivity, and specificity. Initially, the accuracy rate for the varying populations 10, 20, 30, 40, and 50 is measured and the proposed FFO based LSTM classifier obtained the values of 85.494 %, 89.139 %, 90.722 %, 90.817 %, 92.846 % are obtained during the training percentage 80. Similarly, the sensitivity rate of the proposed FFO based LSTM classifier is measured for 80 % of training data and the values are interpreted as 86.803 %, 90.505 %, 92.114 %, 92.209 %, and 94.269 %. Finally, the specificity values for the varying populations are measured and the values are interpreted as 85.040 %, 88.665 %, 90.238 %, 90.333 %, 92.351 %, while using 80 % training data. From the observation, the proposed FFO based LSTM classifier performs well in varying populations.



a)

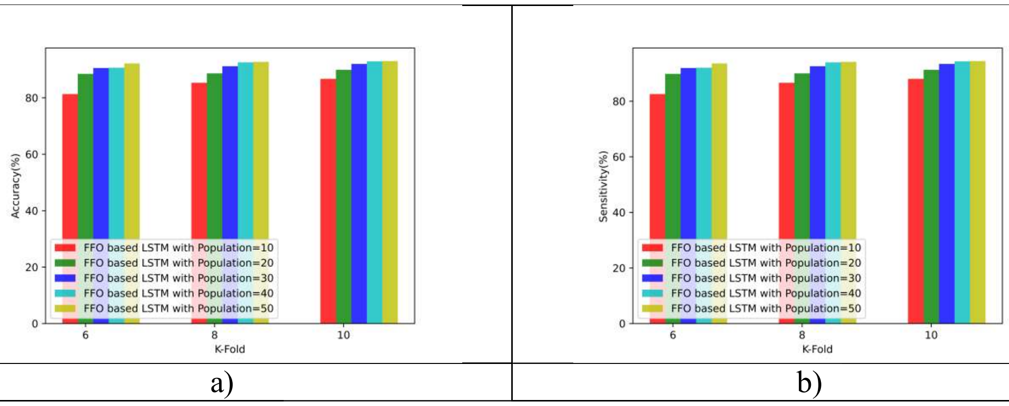
b)

c)

Figure 4. Performance analysis based on dataset-1 concerning training percentage in terms of a) accuracy b) sensitivity and c) specificity

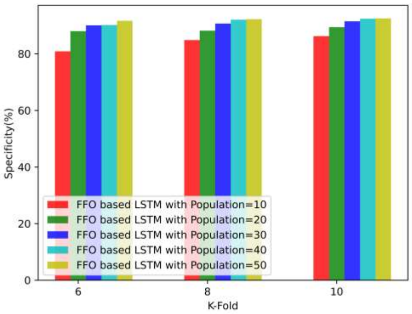
4.3.2 Performance analysis based on dataset-1 concerning the k-fold values

The performance analysis is made based on the k-fold values using dataset 1 for the varying populations 10, 20, 30, 40, 50 and the values are measured for the k-fold values 6, 8, and 10 shown in figures 5 a), b) and c). The accuracy value of the proposed FFO based classifier for varying epochs is measured and enumerated as 86.707 %, 89.906 %, 92.003 %, 92.905 %, and 93.002 % respectively for the k-fold value 10. Similarly, the sensitivity value of the varying epochs are measured and the values obtained for the k-fold 10 are interpreted as 88.035 %, 91.283 %, 93.413 %, 94.330 %, and 94.428 % respectively. Finally, the specificity value for the k-fold value 10 is measured for varying populations and the values obtained are 86.246 %, 89.428 %, 91.514 %, 92.410 %, and 92.506 %, showing the efficacy of the model.



a)

b)

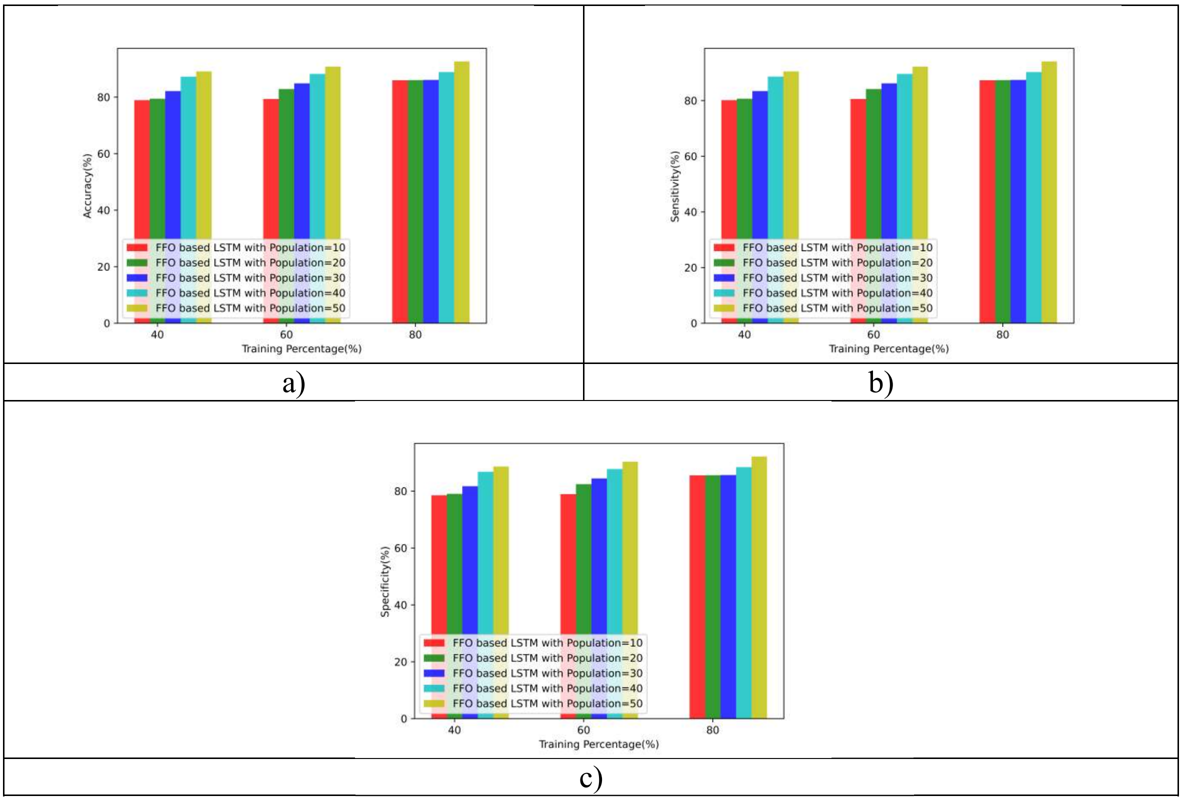


c)

Figure 5. Performance analysis based on dataset-1 concerning k-fold in terms of a) accuracy b) sensitivity and c) specificity

4.3.3 Performance analysis based on dataset-2 concerning the training percentage

The performance analysis is made using dataset-2 and the observation is shown in figures 6 a), b), and c) in terms of accuracy, sensitivity, and specificity. Initially, the accuracy rate for the varying populations 10, 20, 30, 40, and 50 is measured and the proposed FFO based LSTM classifier obtained the values of 85.968 %, 86.020 %, 86.084 %, 88.861 %, 92.633 % are obtained during the training percentage 80. Similarly, the sensitivity rate of the proposed FFO based LSTM classifier is measured for 80 % of training data and the values are interpreted as 87.284 %, 87.339 %, 87.403 %, 90.223 %, and 94.052 %. Finally, the specificity values for the varying populations are measured and the values are enumerated as 85.511 %, 85.561 %, 85.626 %, 88.388 %, 92.140 %, while using 80 % training data. From the observation, the proposed FFO-based LSTM classifier performs well in varying populations.



a)

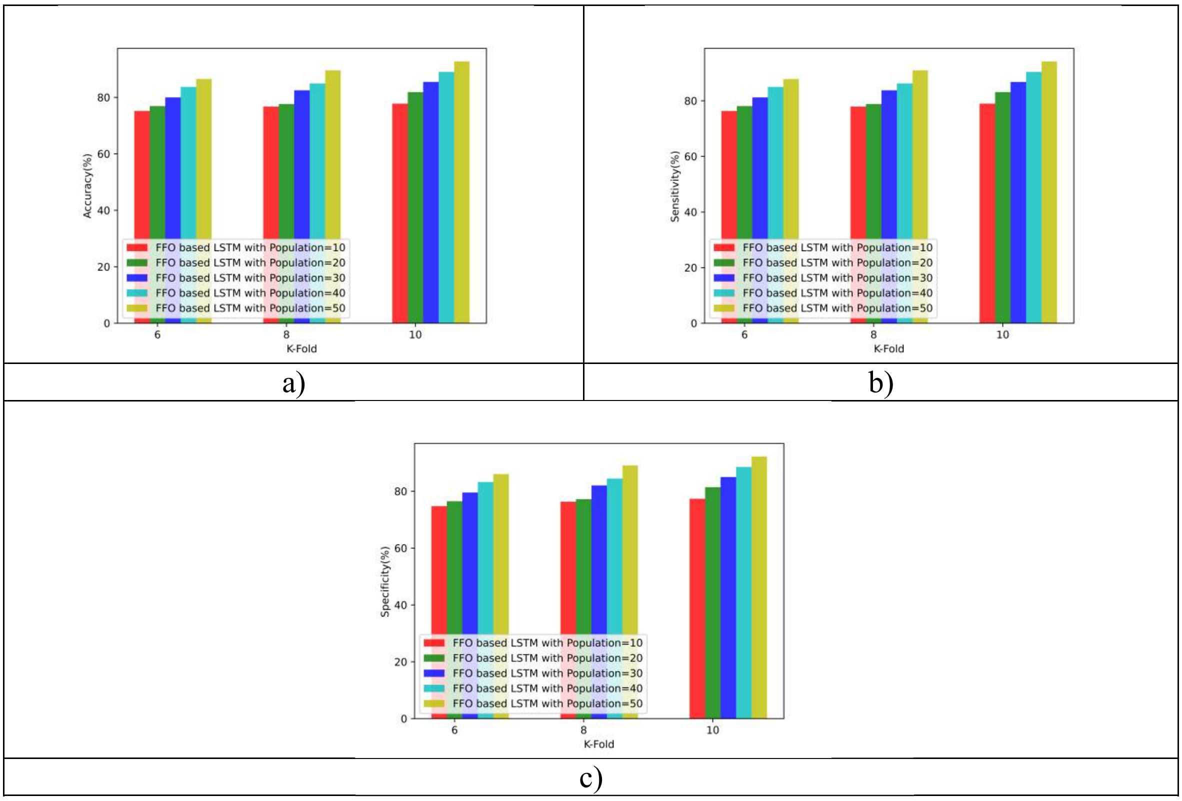
b)

c)

Figure 6. Performance analysis based on dataset-2 concerning training percentage in terms of a) accuracy b) sensitivity and c) specificity

4.3.4 Performance analysis based on dataset-2 concerning the k-fold values

The performance analysis is made based on the k-fold values using the dataset-1 for the varying populations 10, 20, 30, 40, 50 and the values are measured for the k-fold values 6, 8, and 10 shown in figures 7 a), b), and c). The accuracy value of the proposed FFO based classifier for varying epochs is measured and enumerated as 77.773 %, 81.869 %, 85.452 %, 89.001 %, and 92.719 % respectively for the k-fold value 10. Similarly, the sensitivity value of the varying epochs are measured and the values obtained for the k-fold 10 are interpreted as 78.966 %, 83.122 %, 86.761 %, 90.365 %, and 94.140 % respectively. Finally, the specificity value for the k-fold value 10 is measured for varying populations and the values obtained are 77.359 %, 81.434 %, 84.998 %, 88.526 %, and 92.224 %, showing the efficacy of the model.



a)

b)

c)

Figure 7. Performance analysis based on dataset-2 concerning k-fold in terms of a) accuracy b) sensitivity and c) specificity

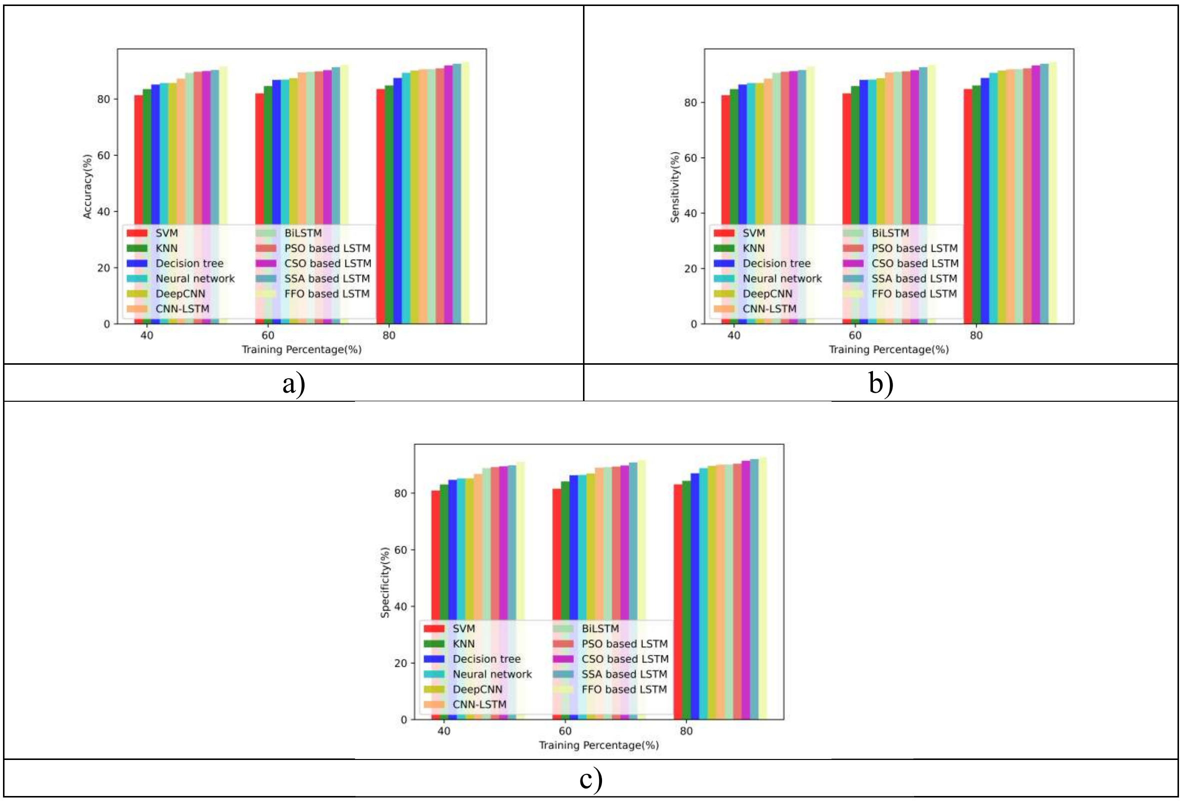
4.4 Comparative analysis

The comparative analysis is performed for measuring the improvement achieved by the proposed FFO based LSTM classifier and the methods used for the comparison are SVM, KNN, Decision tree, NN, Deep CNN, CNN-LSTM, BiLSTM, PSO based LSTM), CSO based LSTM, SSA based LSTM.

4.4.1 Comparative analysis based on training percentage using dataset 1

The comparative analysis is performed for measuring the improvement achieved by the proposed FFO based LSTM classifier and the methods used for the comparison are SVM, KNN, Decision tree, NN, Deep CNN, CNN-LSTM, BiLSTM, PSO based LSTM, CSO based LSTM, SSA based

LSTM. For the simplified view, the improvement rate is interpreted in terms of accuracy, sensitivity, and specificity during the training percentage 80 shown in figures 8 a), b) and c). Initially, the improvement rate obtained by the proposed FFO based LSTM classifier is measured, which attained the rate of 0.644 % improvement while compared with the SSA based LSTM classifier. Similarly, the improvement in terms of sensitivity is measured and the proposed method attained the improvement rate of 0.644% while compared with SSA based LSTM classifier. Finally, the specificity of the proposed FFO based LSTM classifier is measured, which attained an improvement rate of 0.643% while comparing with SSA based LSTM classifier. The values obtained by the proposed method while using dataset 1 at various training percentages are interpreted in table 2.



a)

b)

c)

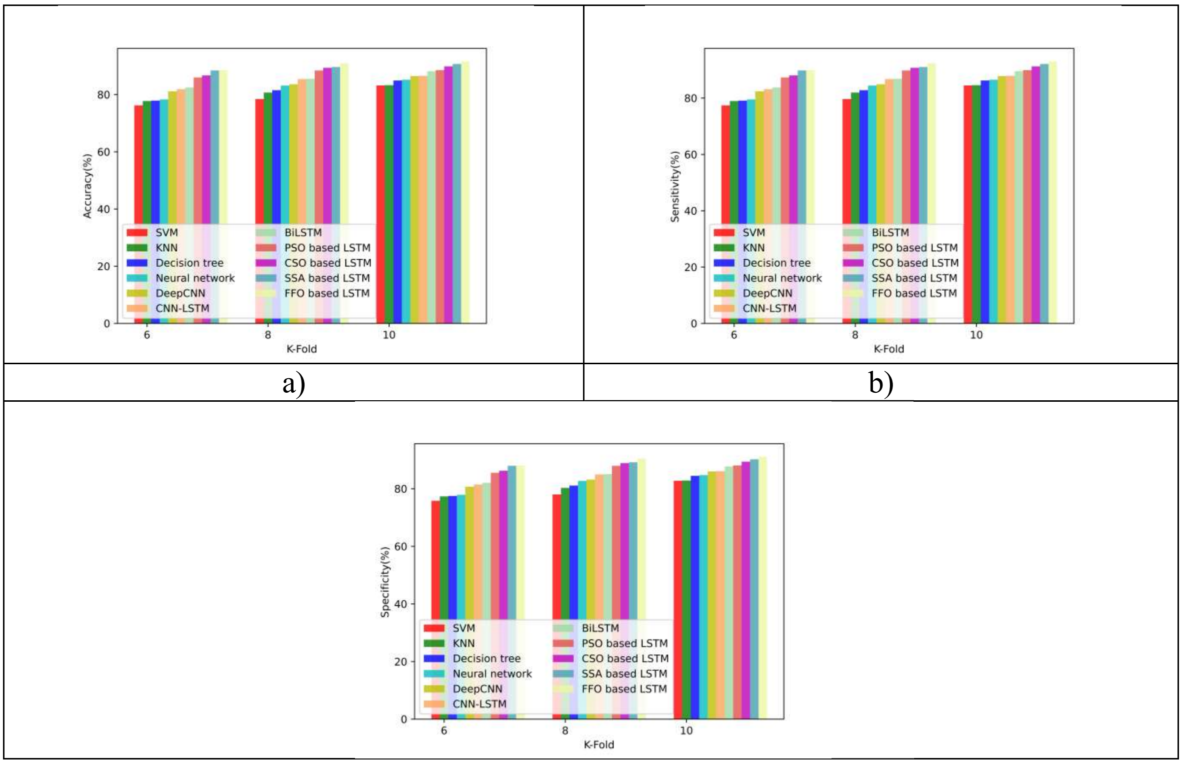
Figure 8. Comparative analysis based on training percentage using dataset-1 in terms of a) accuracy b) sensitivity c) specificity

Table 2: Comparative analysis based on training percentage using dataset 1

Metrics	**Accuracy**			**Sensitivity**			**Specificity**		
Methods/Training percentage	**40**	**60**	**80**	**40**	**60**	**80**	**40**	**60**	**80**
SVM	81.37	82.01	83.55	82.62	83.27	84.83	80.94	81.57	83.10
KNN	83.50	84.59	84.80	84.78	85.89	86.10	83.05	84.13	84.35
Decision tree	85.14	86.79	87.48	86.45	88.12	88.83	84.68	86.33	87.01
NN	85.67	86.91	89.30	86.99	88.24	90.67	85.22	86.44	88.83
Deep CNN	85.69	87.40	90.14	87.01	88.74	91.52	85.23	86.94	89.66
CNN-LSTM	87.25	89.50	90.56	88.59	90.87	91.95	86.79	89.02	90.08
BiLSTM	89.30	89.69	90.61	90.67	91.07	92.00	88.83	89.21	90.13
PSO based LSTM	89.68	89.84	90.89	91.05	91.22	92.28	89.20	89.37	90.40
CSO based LSTM	89.96	90.25	91.96	91.34	91.63	93.37	89.48	89.77	91.46
SSA based LSTM	90.31	91.34	92.56	91.70	92.75	93.98	89.83	90.86	92.06
Proposed	91.55	92.15	93.16	92.95	93.56	94.59	91.06	91.66	92.66

4.4.2 Comparative analysis based on K-fold using dataset 1

The comparative analysis based on the k-fold is performed for the k-fold values 6, 8, and 10 is performed and the improvement rate is measured and is shown in figures 9 a), b), and c). During the K-fold value 10, the proposed FFO based LSTM obtained an improvement rate of 0.933% in terms of accuracy. Similarly, the improvement rate in terms of sensitivity is measured and an improvement rate of 0.939 % is obtained, when compared with SSO based LSTM. Finally, the improvement rate in terms of specificity is measured, which attains the value of 0.928 % during the k-fold value 10. The obtained values shows that the proposed FFO based LSTM perform better than all the existing methods while using the k-fold values.



a)

b)

Figure 9. Comparative analysis based on k-fold using dataset-1 in terms of a) accuracy b) sensitivity c) specificity

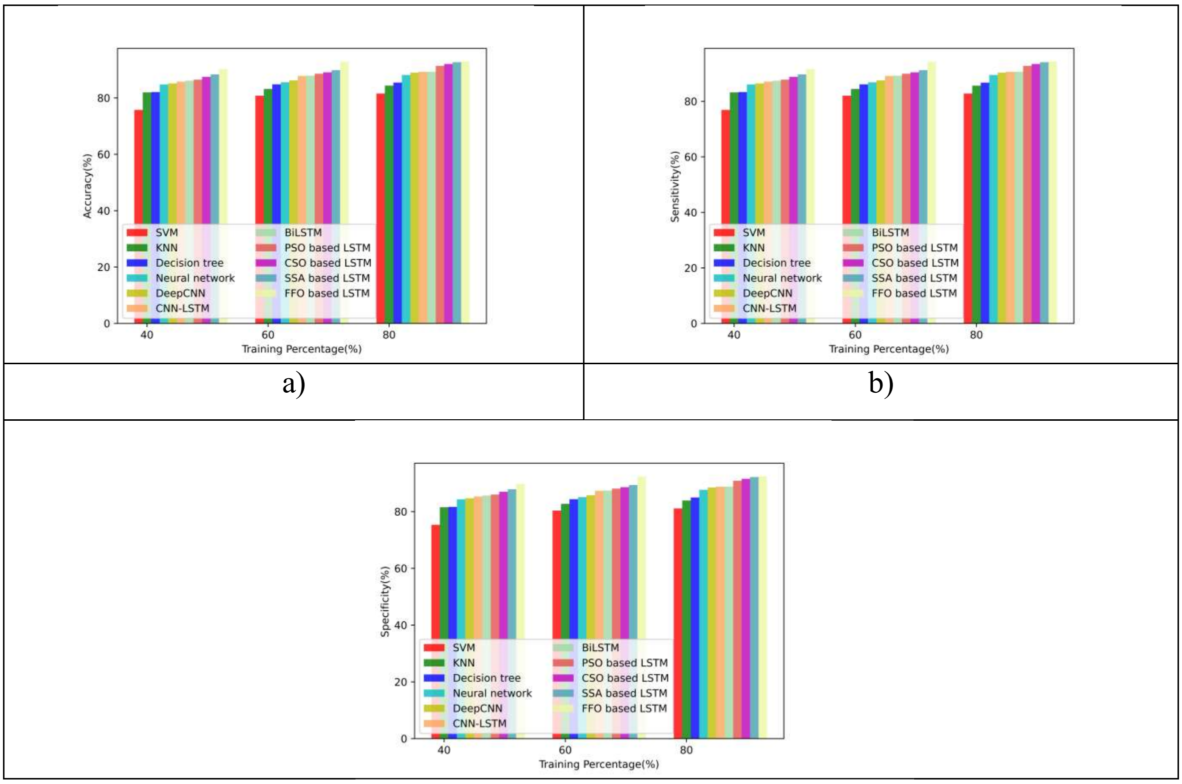
Table 3: Comparative analysis based on training percentage using dataset 1

Metrics	**Accuracy**			**Sensitivity**			**Specificity**		
Methods/k-fold	**6**	**8**	**10**	**6**	**8**	**10**	**6**	**8**	**10**
SVM	76.23	78.45	83.20	77.40	79.66	84.48	75.82	78.03	82.76
KNN	77.75	80.71	83.29	78.94	81.95	84.57	77.33	80.28	82.84
Decision tree	77.91	81.55	84.92	79.11	82.80	86.22	77.49	81.11	84.47
NN	78.32	83.16	85.19	79.52	84.43	86.49	77.90	82.71	84.73
Deep CNN	81.13	83.64	86.46	82.38	84.92	87.79	80.70	83.19	86.00
CNN-LSTM	81.90	85.43	86.55	83.16	86.74	87.88	81.46	84.98	86.09
BiLSTM	82.50	85.52	88.17	83.76	86.83	89.52	82.06	85.07	87.69
PSO based LSTM	85.99	88.44	88.56	87.31	89.80	89.92	85.53	87.97	88.09
CSO based LSTM	86.71	89.36	89.87	88.04	90.73	91.25	86.25	88.88	89.39

SSA based LSTM	88.44	89.64	90.72	89.80	91.02	92.11	87.97	89.16	90.24
Proposed	88.53	90.99	91.57	89.89	92.39	92.98	88.06	90.51	91.08

4.4.3 Comparative analysis based on training percentage using dataset-2

The comparative analysis is performed for measuring the improvement achieved by the proposed FFO based LSTM classifier and the methods used for the comparison are SVM, KNN, Decision tree, NN, Deep CNN, CNN-LSTM, BiLSTM, PSO based LSTM, CSO based LSTM, SSA based LSTM. For the simplified view, the improvement rate is interpreted in terms of accuracy, sensitivity, and specificity during the training percentage 80 shown in figures 10 a), b) and c). Initially, the improvement rate obtained by the proposed FFO based LSTM classifier is measured, which attained the rate of 0.27 % improvement while compared with the SSA based LSTM classifier. Similarly, the improvement in terms of sensitivity is measured and the proposed method attained the improvement rate of 0.274% while compared with SSA based LSTM classifier. Finally, the specificity of the proposed FFO based LSTM classifier is measured, which attained an improvement rate of 0.271 % while comparing with SSA based LSTM classifier. The values obtained by the proposed method while using dataset 1 at various training percentage are interpreted in table 4.



a)

b)

Figure 10. Comparative analysis based on training percentage using dataset-2 in terms of a) accuracy b) sensitivity c) specificity

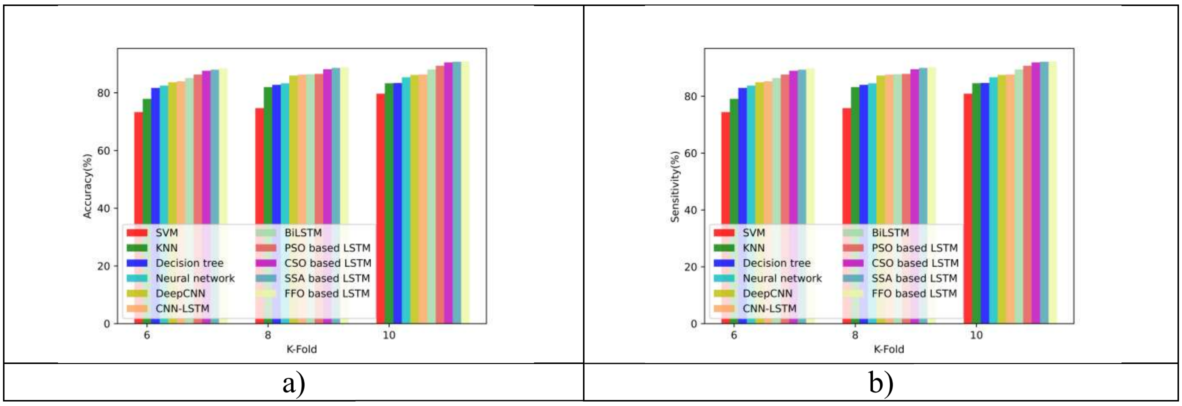
Table 4: Comparative analysis based on training percentage using dataset-2

Metrics	**Accuracy**			**Sensitivity**			**Specificity**		
Methods/k-fold	**40**	**60**	**80**	**40**	**60**	**80**	**40**	**60**	**80**
SVM	75.75	80.78	81.57	76.91	82.03	82.81	75.34	80.35	81.13
KNN	81.96	83.17	84.35	83.21	84.44	85.64	81.52	82.72	83.90
Decision tree	82.09	84.81	85.42	83.35	86.11	86.73	81.65	84.36	84.97
NN	84.79	85.53	88.14	86.09	86.85	89.50	84.34	85.08	87.67
Deep CNN	85.15	86.21	88.98	86.45	87.53	90.35	84.69	85.75	88.51
CNN-LSTM	85.79	87.81	89.24	87.10	89.15	90.61	85.33	87.34	88.77
BiLSTM	86.11	87.90	89.28	87.43	89.24	90.65	85.65	87.43	88.80
PSO based LSTM	86.47	88.58	91.36	87.80	89.94	92.76	86.00	88.11	90.87

CSO based LSTM	87.46	89.08	92.05	88.81	90.45	93.47	86.99	88.61	91.56
SSA based LSTM	88.33	89.86	92.68	89.68	91.24	94.10	87.86	89.39	92.18
Proposed	90.25	92.77	92.93	91.63	94.20	94.35	89.77	92.28	92.43

4.4.4 Comparative analysis based on K-fold using dataset 2

The comparative analysis based on the k-fold is performed for the k-fold values 6, 8, and 10 is performed and the improvement rate is measured and is shown in figures 11 a), b), and c). During the K-fold value 10, the proposed FFO based LSTM obtained an improvement rate of 0.104 % in terms of accuracy. Similarly, the improvement rate in terms of sensitivity is measured and an improvement rate of 0.103 % is obtained, when compared with SSO based LSTM. Finally, the improvement rate in terms of specificity is measured, which attains the value of 0.104 %, during the k-fold value 10. The obtained values show that the proposed FFO based LSTM performs better than all the existing methods while using the k-fold values.



a) b)

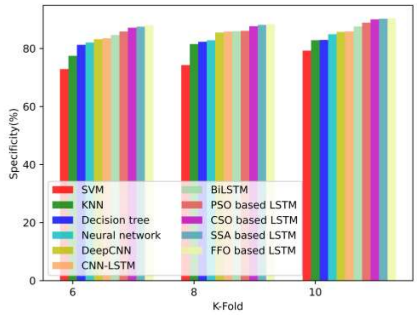


Figure 11. Comparative analysis based on k-fold using dataset-2 in terms of a) accuracy b) sensitivity c) specificity

Table 5: Comparative analysis based on k-fold using dataset-2

Metrics	**Accuracy**			**Sensitivity**			**Specificity**		
Methods/k-fold	**6**	**8**	**10**	**6**	**8**	**10**	**6**	**8**	**10**
SVM	73.29	74.69	79.69	74.42	75.84	80.92	72.90	74.29	79.27
KNN	77.89	81.97	83.29	79.09	83.23	84.57	77.47	81.53	82.85
Decision tree	81.68	82.75	83.38	82.93	84.01	84.66	81.25	82.31	82.93
NN	82.48	83.27	85.35	83.74	84.55	86.66	82.03	82.83	84.90
Deep CNN	83.62	85.95	86.18	84.90	87.27	87.50	83.18	85.49	85.72
CNN-LSTM	83.97	86.27	86.31	85.26	87.59	87.63	83.52	85.81	85.85
BiLSTM	85.10	86.45	88.06	86.41	87.77	89.41	84.64	85.99	87.59
PSO based LSTM	86.33	86.54	89.35	87.65	87.86	90.72	85.86	86.08	88.88
CSO based LSTM	87.63	88.15	90.54	88.97	89.50	91.93	87.16	87.68	90.06
SSA based LSTM	87.99	88.61	90.73	89.34	89.97	92.12	87.53	88.14	90.25
Proposed	88.53	88.83	90.82	89.89	90.20	92.22	88.05	88.36	90.34

4.5 Comparative discussion

The comparative discussion is performed to prove the superiority of the proposed FFO based LSTM over the state of art methods. The proposed method obtained better accuracy of 93.16 %, a sensitivity of 94.59 %, and specificity of 92.66 % while concerning the training percentage using dataset 1. Similarly, concerning k-fold values, the value of 91.573 % in terms of accuracy, 92.981 % in terms of specificity, and 91.082 % in terms of specificity are obtained while using dataset 1. Then the value of the metrics for dataset-2 is measured and concerning the training percentage the values of 92.927 % in terms of accuracy, 94.354 % in terms of sensitivity, and 92.429 % in terms of specificity are obtained. While considering the k-fold values the accuracy of 90.824 %, the sensitivity of 92.217 %, and specificity of 90.339 %, which is more efficient. The values obtained using various datasets and metrics values are interpreted in table 6.

Metrics/ Methods	**Training percentage**						**K-fold**					
	Dataset 1			**Dataset-2**			**Dataset-1**			**Dataset-2**		
	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity
SVM	83.20	84.83	83.10	81.57	82.81	81.13	83.20	84.48	82.76	79.69	80.92	79.27
KNN	83.29	86.10	84.35	84.35	85.64	83.90	83.29	84.57	82.84	83.29	84.57	82.85
Decision tree	84.92	88.83	87.01	85.42	86.73	84.97	84.92	86.22	84.47	83.38	84.66	82.93
NN	85.19	90.67	88.83	88.14	89.50	87.67	85.19	86.49	84.73	85.35	86.66	84.90
Deep CNN	86.46	91.52	89.66	88.98	90.35	88.51	86.46	87.79	86.00	86.18	87.50	85.72
CNN-LSTM	86.55	91.95	90.08	89.24	90.61	88.77	86.55	87.88	86.09	86.31	87.63	85.85
BiLSTM	88.17	92.00	90.13	89.28	90.65	88.80	88.17	89.52	87.69	88.06	89.41	87.59
PSO based LSTM	88.56	92.28	90.40	91.36	92.76	90.87	88.56	89.92	88.09	89.35	90.72	88.88
CSO based LSTM	89.87	93.37	91.46	92.05	93.47	91.56	89.87	91.25	89.39	90.54	91.93	90.06
SSA based LSTM	90.72	93.98	92.06	92.68	94.10	92.18	90.72	92.11	90.24	90.73	92.12	90.25

Proposed	91.57	94.59	92.66	92.93	94.35	92.43	91.57	92.98	91.08	90.82	92.22	90.34

6. Conclusion

Phishing detection technologies play a crucial part in ensuring that users have a secure online experience by protecting them from being the victims of online fraud, giving personal information to an attacker, and so on. Detection of phishing websites using the FFO based LSTM classifier is performed in this research to prevent the scams occurs in the emails. The LSTM provides large number of parameters and the necessary parameters are selected and optimized using the FFO algorithm, which helps in the reducing the computational complexity of the system and boosts the performance. A deeper analysis of the network is performed by the LSTM classifier, which helps in improving the performance of phishing detection. The imbalance problem also resolved in this research using SMOTE, which reduces the uneven distribution of the data. The accuracy, sensitivity, and specificity of the metrics are used to illustrate the significance of the research, and the proposed FFO-based LSTM classifier achieved the remarkable values of 93.16%, 94.59%, and 92.66%, which shows the efficiency of the research.

Reference

[1] Shan Wang., Khan, S., Xu, C., Nazir, S. and Hafeez, A., "Deep learning-based efficient model development for phishing detection using random forest and BLSTM classifiers", Complexity,vol.2020, pp.1-7, 2020.

[2] Mohanta, B.K., Jena, D., Satapathy, U. and Patnaik, S., "Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology", Internet of Things, vol.11, pp.100227, 2020.

[3] Gunikhan Sonowala, and K S Kuppusamy. and Kuppusamy, K.S., "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] AlEroud, A. and Karabatis, G., "Bypassing detection of URL-based phishing attacks using generative adversarial deep neural networks", In Proceedings of the Sixth International Workshop on Security and Privacy Analytics, pp. 53-60, 2020.

[5] Victor E. Adeyemo., Balogun, A.O., Mojeed, H.A., Akande, N.O. and Adewole, K.S., "Ensemble-Based Logistic Model Trees for Website Phishing Detection", Advances in Cyber Security, pp. 627-641, 2020.

[6] US state department hack, "https://securityintelligence.com/us-state-department-hack-has-major-security-implications/".

[7]Researcher uncover RSA phishing attack," https://www.wired.com/2011/08/how-rsa-got-hacked/".

[8]Phishing test as real attack, "https://www.wired.com/story/dnc-phishing-test-votebuilder/".

[9] Qi Li., Cheng, M., Wang, J. and Sun, B.,"LSTM based phishing detection for big email data", IEEE Transactions on Big Data, 2020.

[10] Liqun Yang., Zhang, J., Wang, X., Li, Z., Li, Z. and He, Y., "An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features", Expert Systems with Applications, vol.165, pp.113863, 2021.

[11] Erzhou Zhu., Ju, Y., Chen, Z., Liu, F. and Fang, X., "DTOF-ANN: An artificial neural network phishing detection model based on decision tree and optimal features", Applied Soft Computing, vol.95, pp.106505, 2020.

[12] Meng, Y. and Kwok, L.F., "Adaptive blacklist-based packet filter with a statistic-based approach in network intrusion detection", Journal of Network and Computer Applications, vol.39, pp.83-92, 2014.

[13] Moustafa, N., Misra, G. and Slay, J., "Generalized outlier gaussian mixture technique based on automated association features for simulating and detecting web application attacks", IEEE Transactions on Sustainable Computing, 2018.

[14] Tan, C.L., Chiew, K.L., Yong, K.S., Abdullah, J. and Sebastian, Y.,"A graph-theoretic approach for the detection of phishing webpages", Computers & Security, vol.95, pp.101793, 2020.

[15] Adewole, K.S., Akintola, A.G., Salihu, S.A., Faruk, N. and Jimoh, R.G.,"Hybrid rule-based model for phishing URLs detection", In proceedings of International Conference for Emerging Technologies in Computing, pp. 119-135, 2019.

[16] Zamir, A., Khan, H.U., Iqbal, T., Yousaf, N., Aslam, F., Anjum, A. and Hamdani, M., "Phishing web site detection using diverse machine learning algorithms", The Electronic Library, 2020.

[17] Ghafir, I. and Prenosil, V.,"Blacklist-based malicious ip traffic detection", In proceedings of 2015 Global Conference on Communication Technologies (GCCT), pp. 229-233, 2015.

[18] Hong, J., "The state of phishing attacks", Communications of the ACM, vol.55, no.1, pp.74-81, 2012.

[19] Chu, S.C., Tsai, P.W. and Pan, J.S., "Cat swarm optimization", In proceedings of Pacific Rim international conference on artificial intelligence, pp. 854-858, 2006.

[20] Xue, J. and Shen, B., "A novel swarm intelligence optimization approach: sparrow search algorithm", Systems Science & Control Engineering, vol.8, no.1, pp.22-34, 2020.

[21] Lakshmi, L., Reddy, M.P., Santhaiah, C. and Reddy, U.J., "Smart Phishing Detection in Web Pages using Supervised Deep Learning Classification and Optimization Technique ADAM", Wireless Personal Communications, vol.118, no.4, pp.3549-3564, 2021.

[22] Ammar Odeh., Keshta, I. and Abdelfattah, E., "PHIBOOST-a novel phishing detection model using Adaptive boosting approach", Jordanian Journal of Computers and Information Technology (JJCIT), vol.7, no.01, 2021.

[23] Phishing Website Detection by Machine Learning Techniques dataset, https://github.com/shreyagopal/Phishing-Website-Detection-by-Machine-Learning-Techniques, Accessed on November 2021.

[24] Hannousse, Abdelhakim; Yahiouche, Salima (2021), “Web page phishing detection”, Mendeley Data, V3, doi: 10.17632/c2gw7fy2j4.3 Accessed on November 2021.

[25] Binu, D., and B. S. Kariyappa. "RideNN: A new rider optimization algorithm-based neural network for fault diagnosis in analog circuits." IEEE Transactions on Instrumentation and Measurement 68, no. 1 (2018): 2-26.