IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-12, Dec 2019 E-ISSN: 2347-2693

Ensembling of Stacked Denoise Autoencoder for Phishing Attack Detection

K. Sumathi[1*], V. Sujatha[2]

[1,2]Dept. of Computer Application, CMS College of Science and Commerce, Chinnavedampatty, Coimbatore, Tamilnadu, India

[*]*Corresponding Author: nksmba@rediffmail.com, Tel.: +91-98994200641*

DOI: https://doi.org/10.26438/ijcse/v7i12.115121 | Available online at: www.ijcseonline.org

Accepted: 19/Dec/2019, Published: 31/Dec/2019

***Abstract*—**Phishing is one of the most severe threats to internet security. It utilizes spotted websites to rob users' passwords and online identities. Generally, phishers use spotted emails or instant messages to attract users to phishing websites. In order to detect phishing attacks in the network, Deep Neural Network (DNN) was introduced. However, the computational complexity of DNN-based phishing attack detection is high because of using irrelevant and redundant features in DNN. So, DNN with Stacked Denoise AutoEncoder (DNN-SDAE) was proposed which reconstructed input features by removing irrelevant and redundant features. Then, the softmax activation function was processed the reconstructed features detect the phishing attack. In this paper, DNN with Ensembling SDAE (DNN-ESDAE) is proposed to reduce the complexity of SDAE and enhance the phishing attack detection accuracy. Initially, Uniform Resource Locator (URL)-based features, Hyper-Text Markup Language (HTML)-based features and domain-based features are extracted by using feature extractor. Then, individual type of features is processed in different SDAE which reconstruct input features. After the ensembling of three SDAE using negative correlation learning, the best selective ensembling is chosen using Shuffled Frog Leaping Optimization Algorithm (SFLOA). Finally, majority voting is employed to combine the results of three SDAE. The experiment is conducted to prove the effectiveness of DNN-ESDAE in terms of accuracy, precision, recall, and f-measure.

***Keywords*—**Phishing attack detection, Deep Neural Network, Ensembling Stacked Denoise AutoEncoder, Shuffled Frog Leaping Optimization Algorithm, majority voting

I. INTRODUCTION

Phishing [1] is a type of frustration that involves trying to obtain personal or sensitive information using social engineering. Phishing [2] is usually performed through email where an assailant acts as a trustworthy or reliable source to induce the receiver to click on a link or to open an attachment within an email. Industry reports also concluded that phishing emails are a major threat to organizational information security for employees. Furthermore, research has shown that phishing emails often use principles of social influence to convince the user to comply. In order to reduce this threat and to protect personal, sensitive or organizational information, an efficient technique for phishing email detection is more required.

For efficient phishing attack detection, good quality of training data is required which obtained by using deep learning technique [3]. A machine learning technique [4] called Deep Neural Network (DNN) [5] was introduced for phishing email detection. A feature extractor was used to extract URL-based features, HTML-based features and domain-based features were extracted and those features were processed in DNN for phishing attack detection. The complexity of DNN was high since it processed the irrelevant, redundant and noisy features (missing data). So, DNN with Stacked Denoise Auto Encoder (SDAE) [6] was proposed where SDAE was used to reconstruct an input feature vector for phishing attack detection. The reconstructed feature vectors were processed in the DNN to classify the emails as a phishing emails and legitimate emails.

In this paper, DNN with Ensembling SDAE (DNN-ESDAE) is proposed to reduce the computational complexity of reconstruction of input feature vectors and to enhance the accuracy of phishing attack detection. In DNN-ESDAE, three SDAE are used to separately reconstruct the URL-based features, HTML-based features, and domain-based features. Then, negative correlation learning is used to fine-tune the softmax classifier. Finally, the phishing detection results of three SDAE are ensemble using Shuffled Frog Leaping Optimization Algorithm (SFLOA) and majority voting. Thus, the computational complexity of SDAE is

reduced by handling the URL-based features, HTML-based features, and domain-based features separately and phishing detection accuracy is improved by using the ensembling method.

The main contributions of this paper are listed as follows:

1. To reduce the computational complexity for the reconstruction of input feature vectors.
2. To enhance the phishing detection accuracy by ensembling the results of SDAE.

The rest of the article is structured as follows: Section II provides the previous researches related to phishing detection techniques. Section III explains the proposed DNN with Ensembling SDAE for phishing detection in brief. Section IV compares the performance of the proposed method with the existing method and Section V concludes the research work.

II. RELATED WORK

Hamid & Abawajy [7] proposed a hybrid feature selection method for phishing email detection. According to the combination of behavior-based phishing detection and content-based phishing detection approach, the hybrid feature selection method was processed. It used to determine the knowledge about attackers which was extracted from an email header. It analyzed the sender email and message-ID tag to mine attacker's behavior. However, the hybrid feature selection method does not work on graphical form as some attackers bypass the content-based approach.

Montazer & ArabYarmohammadi [8] introduced a fuzzy-rough hybrid system for the detection of phishing attacks in Iranian e-banking. This system identified the influential features of phishing that best fit the Iranian bank sites. Then, rough set theory was applied to select the most discriminative features and those features were used in a fuzzy expert system for phishing detection. However, the membership function of a fuzzy expert system greatly influences the efficiency of phishing detection.

Sonowal & Kuppusamy [9] proposed a multilayer model named as Phishing Detection using Multi-filter Approach (PhiDMA) for phishing detection. The multilayer model was comprised of five different layers. The first layer handled the appropriate matching of the current URL with the whitelist's URL. The second layer verified URL's contents and it extracted the distinct features. The third layer verified the URL using a search engine result's list. The fourth layer measured the similarity percentage of the current URL with search engine result URLs and the fifth layer dealt with accessibility score similarity. More features should be included to accomplish better performance of PhiDMA.

Smadi et al. [10] proposed a novel framework for phishing attack detection. This framework was a combination of neural network with reinforcement learning. Feature Evaluation and Reduction (FEaR) was developed to use the new behavior and to prioritize the selected list of features. After that Dynamic Evolving Neural Network using Reinforcement Learning (DENNuRL) was developed for phishing email detection which used the features retained by the FEaR method. More datasets could be included in the offline dataset to increase the richness of this framework.

Yang et al. [11] proposed a Multi-dimensional Phishing Detection (MFPD) approach for phishing detection. Initially, character sequence features of the URL were extracted and processed by deep learning. This process doesn't require any prior knowledge about phishing and third-party assistance. After that, webpage text features, URL statistical features, quick classification result, and webpage code features were integrated to reduce the phishing attack detection time. Based on a threshold value, the phishing attacks were detected. However, the threshold value greatly influences the accuracy of phishing attack detection.

Chatterjee & Namin [12] proposed a novel approach based on deep reinforcement learning for phishing website detection. In this approach, an agent in the reinforcement learning learned the value function from the URL to perform a classification task. Then, a deep neural network was implemented to map the sequential decision-making process for the classification of URL as phishing websites or legitimate websites. However, this approach needs improvement in terms of accuracy.

III. METHODOLOGY

In this section, the DNN-ESDAE for phishing email detection is described in detail. Initially, a feature extractor is used which processed URLs and web-based code to extract URL-based features [13], HTML-based features [14] and domain-based features [15]. The extracted three different types of features are processed by different SDAE separately to reconstruct the features. Then, softmax classifier fine-tunes by non-negative learning and the result of each classifier is ensemble by frog leaping and majority voting technique. The ensembling of DNN-SDAE improves the performance of phishing detection compared to that of single DNN-SDAE.

Initially in DNN-ESDAE, the extracted URL-based features are processed by DNN-SDAE1, HTML-based features are processed in DNN-SDAE2 and domain-based features are processed in DNN-SDAE3. The DNN-SDAE consists of encoder (i.e., hidden layer) and decoder (i.e., output layer). The encoder maps the features from high-dimensional space into codes within a low-dimensional space. The decoder

reconstructs the features from the corresponding codes. For the training features x_i, a non-linear mapping is used in encoder to convert the input vector x into a hidden representation. The encoder functions of DNN-SDAE1, DNN-SDAE2 and DNN-SDAE3 which is given as follows:

$$h_{URL} = \sigma(w_1 x_{URL} + b_1) \quad (1)$$

$$h_{HTML} = \sigma(w_1 x_{HTML} + b_1) \quad (2)$$

$$h_{domain} = \sigma(w_1 x_{domain} + b_1) \quad (3)$$

In the above equations, $x_{URL} \in URL\_features$, $x_{HTML} \in HTML\_features$, $x_{domain} \in domain\_features$, h_{URL} is the hidden representation of URL features, h_{HTML} is the hidden representation of HTML features, h_{domain} is the hidden representation of domain features,σ is the activation function, w_i is the weight value and b_i is the bias term. After the encoder process, the decoder maps hidden representation back to the original representation which is given as follows:

$$z_{URL} = \sigma(w_2 x_{URL} + b_2) \quad (4)$$

$$z_{HTML} = \sigma(w_2 x_{HTML} + b_2) \quad (5)$$

$$z_{domain} = \sigma(w_2 x_{domain} + b_2) \quad (6)$$

The DNN-SDAE training intends to fine-tune parameter set $\delta = \{w_1, b_1, w_2, b_2\}$ to reduce the reconstruction error between z and x. Mean Average Error (MSE) is used to measure average reconstruction error which is calculated as,

$$J_{MSE\_URL}(\delta) = \frac{1}{m\_URL}\sum_{i=1}^{m}\left(\frac{1}{2}\left(z_{URL(i)} - x_{URL(i)}\right)^2\right) = \frac{1}{m\_URL}\sum_{i=1}^{m}\left(\frac{1}{2}\left(f_{w,b}\left(x_{URL(i)}\right) - x_{URL(i)}\right)^2\right) \quad (7)$$

$$J_{MSE\_URL}(\delta) = \frac{1}{m\_HTML}\sum_{i=1}^{m}\left(\frac{1}{2}\left(z_{HTML(i)} - x_{HTML(i)}\right)^2\right) = \frac{1}{m\_HTML}\sum_{i=1}^{m}\left(\frac{1}{2}\left(f_{w,b}\left(x_{HTML(i)}\right) - x_{HTML(i)}\right)^2\right) \quad (8)$$

$$J_{MSE\_domain}(\delta) = \frac{1}{m\_domain}\sum_{i=1}^{m}\left(\frac{1}{2}\left(z_{domain(i)} - x_{domain(i)}\right)^2\right) = \frac{1}{m\_domain}\sum_{i=1}^{m}\left(\frac{1}{2}\left(f_{w,b}\left(x_{domain(i)}\right) - x_{domain(i)}\right)^2\right) \quad (9)$$

The DNN-SDAE1, DNN-SDAE2 and DNN-SDAE3 are ensembling as DNN-ESDAE which improve the performance of phishing attack detection model.

A. Extraction of important feature representation using Negative correlation learning

The DNN-ESDAE used negative correlation learning to extract important feature representations. A set of different subspaces chosen by negative correlation learning resulted in diversity is generated when the used features in the training data are perturbed. Negative correlation learning emphasizes interaction among the features in the ensemble to increase the diversity among them, by introducing penalty terms in the objective function during training. Negative correlation learning is a mean result of three DNN-SDAE is given as follows:

$$Fea(n) = \frac{1}{l}\sum_{i=1}^{l} Fea_i(n) \quad (10)$$

Equation (10), l is the number of features in the ensemble, $Fea_i(n)$ is the reconstructed features of the ith DNN-SDAE on the nth training sample and $Fea_i(n)$ ensembling features of the nth training sample. Negative correlation learning introduces a correlation penalty term into the objective of each DNN-SDAE so that all DNN-SDAE is constructed simultaneously and interactively on the same training dataset. When the nth training pattern is available, the ith DNN-SDAE is fine-tuned to minimize the following error function:

$$E_i(n) = \frac{1}{2}\left(Fea_i(n) - d(n)\right)^2 + \gamma p_i(n) \quad (11)$$

Equation (11), γ is the control parameter which is ranges from 0 to 1. It is used to tradeoff between MSE and the penalty term $p_i(n)$. It is defined as follows:

$$p_i(n) = \left(Fea_i(n) - Fea(n)\right)\sum_{j\neq i}\left(Fea_j(n) - Fea(n)\right) \quad (12)$$

The above penalty term explicitly encourages the ith DNN-SDAE to be negatively correlated with the rest SDAE in the ensemble. The partial derivation of $E_i(n)$, concerning the output of the ith DNN-SDAE on the nth training sample is,

$$\frac{\partial E_i(n)}{\partial Fea_i(n)} = \big(Fea_i(n) - d(n)\big) + \mu \frac{\partial p_i(n)}{\partial Fea_i(n)}$$
$$= (1-\mu)\big(Fea_i(n) - d(n)\big) + \mu\big(Fea(n) - d(n)\big) \quad (13)$$

By using (13), the weight of all three DNN-SDAE is updated by using back propagation.

B. Selective ensemble learning for phishing attack detection

Instead of combining all the three DNN-SDAE, selective DNN-SDAE is combined to reduce the computational complexity of phishing attack detection. The selective DNN-SDAE is an ensemble based on Shuffled Frog Leaping Optimization Algorithm (SFLOA). It generates an optimal subset of DNN-SDAE. The selection problem of DNN-SDAE is formulated as the optimization problem which is given as follows:

$$min_w J(w), w = [w_1, w_2, w_3]$$
$$Subject\ to\ num(w_i > 0) = P \quad (14)$$

Equation (14), $J(w)$ is the objective function of individual DNN-SDAE. The DNN-ESDAE used the following objective function to select the P DNN-SDAEs. This objective function considers accuracy and diversity of the ensemble which is defined as follows:

$$F(\mathcal{X}) = accuracy + \beta \times diversity$$

$$= \frac{1}{N}\sum_{n=1}^{N}\big(Fea_i(n) - y(n)\big)^2 - E\left[\left(\frac{x-\mu}{\sigma}\right)^4\right] \quad (15)$$

Equation (15), β is the tradeoff between accuracy (MSE) and diversity (kurtosis) of DNN classifier.

C. Shuffled Frog Leaping Optimization Algorithm

The best ensemble of DNN-SDAE is selected by SFLOA which performs a heuristic search based on the evolution of particle called memes. It carried by several frogs that perform a global exchange of information among the population. SFLOA tries to imitate the search for food by a group of frogs that exchange information among themselves. Each frog has a specific location in the search space (S^i). Its vector denotes a meme with different memotypes as decision variables D. Each memotype identities the discrete value of each decision variable.

$$S^i = \{S_1^i, S_2^i, \dots S_D^i\} \quad (16)$$

The global exchange of information between the memes has a probabilistic component.

Consider initial population P which is randomly generated by SFLOA. Each of the frogs with different ensembling of DNN-SDAE in the initial population is sorted based on the objective function in (15). The initial population is split into h number of memeplexes. Each memeplex contains q frogs and can be assumed as a different culture in which a local search is performed. Then, frogs are forwarded to different memeplexes based on their objective function. Each memeplex is split into sub-memeplex that denotes the number of frogs entering memetic evolution. Frogs exchange information within each sub-memeplex, thus best informs to the worst which evolves in a process called an evolutionary leap. In this process, only the frog with the worst solution in each iteration is updated which is given as follows:

$$L_i = \delta \times C \times (S_{best} - S_{worst}) \quad (17)$$

$$S_{w,1} = S_{w,0} + L_i(L_{max} \geq L_i \geq -L_{max}) \quad (18)$$

Equation (17) and (18), the best solution for each memeplex is represented as S_{best}, the worst solution for each memeplex is represented as S_{worst}, L_i is the change in frog location, $\delta \in (0,1)$ is the random number, $S_{w,0}$ denotes the current location of frog i, $S_{w,1}$ denotes the new location of frog i and L_{max} is the maximum allowed a change in a frog's location. If the evolution generates a better frog, it replaces the worst frog otherwise S_{best} is replaced by global fitness S_{glob} in (17) and the process is continued. If the objective function of the new frog is not better than objective function of S_{worst}, then a new frog is generated randomly for replacing the worst frog. This process is continued for a particular number of iterations within each sub-memeplex. Thus, the local search in each sub-memeplex is finished and the sub-memeplexes are returned to memeplexes.

The memeplexes are dissolved and the shuffling process is initiated where frogs are mixed again based on their objective function and re-sorted into new memeplexes. According to this, a generation is completed. Finally, the SFLOA has the ability to evolve a random initial to the global minimum. The leaping and shuffling process are continued until the convergence is satisfied. Thus, the optimal ensembling of DNN-SDAE is obtained based on SFLOA. Finally, the majority voting is used to combine the output (i.e., phishing attack detection) of the final ensemble.

IMMA-ESDAE for phishing detection algorithm

Step 1: Get the URL-based features, HTML-based features and domain-based features from feature extractor.

Step 2: Process URL-based features, HTML-based features and domain-based features in DNN-SDAE1, DNN-SDAE2 and DNN-SDAE3 correspondingly.

Step 3: Negative correlation learning is implemented in each DNN-SDAE to extract important feature representation.
Step 4: Selective ensemble is implemented using SFLOA to select the SDAE for ensembling.
Step 5: Majority voting method is used to combine outputs of ensembling SDAE.

IV. RESULT AND DISCUSSION

The efficiency of proposed DNN-ESDAE is compared against DNN-SDAE in terms of accuracy, precision, recall and f-measure. For the experimental purpose, Ham, Phishing Corpus and Phishload datasets [4] are used.

A. Accuracy

It measures the overall rate of correctly detected legitimate and phishing URLs.

$$Accuracy = \frac{True\ Positive\ (TP)+True\ Negative\ (TN)}{TP+TN+False\ Positive\ (FP)+False\ Negative\ (FN)} \quad (19)$$

where, TP is the percentage of phishing URLs in the training dataset that is correctly classified as phishing URLs, TN is the percentage of legitimate URLs in the training dataset that is correctly classified as legitimate URLs, FP is the percentage of legitimate URLs that are incorrectly classified as phishing URLs and FN is the percentage of phishing URLs that are incorrectly classified as legitimate URLs.
Table 1 shows the accuracy of DNN-SDAE and DNN-ESDAE on Ham, Phishing Corpus and Phishload datasets.

Table 1. Evaluation of Accuracy

Datasets	DNN-SDAE	DNN-ESDAE
Ham	0.92	0.94
Phishing Corpus	0.94	0.957
Phishload	0.914	0.939

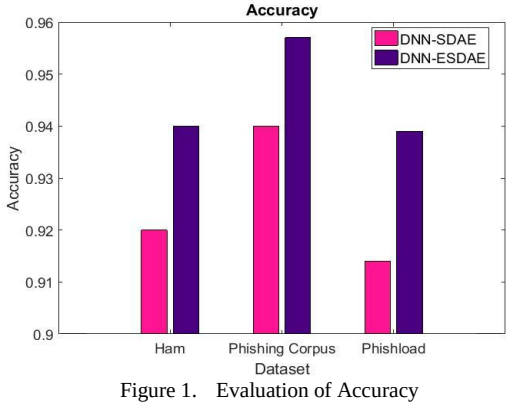

Figure 1. Evaluation of Accuracy

The accuracy of DNN-SDAE and DNN-ESDAE on three different datasets is shown in Figure 1. The accuracy of DNN-ESDAE is 2.17%, 1.81%, and 2.74% greater than DNN-SDAE on Ham, Phishing Corpus, and Phishload datasets respectively. From this analysis, it is known that the proposed DNN-ESDAE has high accuracy than DNN-SDAE for phishing attack detection.

B. Precision

It measures the exactness of the DNN i.e., what percentage of URLs that the classifier labeled as phishing URLs and it is calculated as,

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

Table 2 shows the precision of DNN-SDAE and DNN-ESDAE on Ham, Phishing Corpus and Phishload datasets.

Table 2. Evaluation of Precision

Datasets	DNN-SDAE	DNN-ESDAE
Ham	0.91	0.928
Phishing Corpus	0.916	0.93
Phishload	0.90	0.92

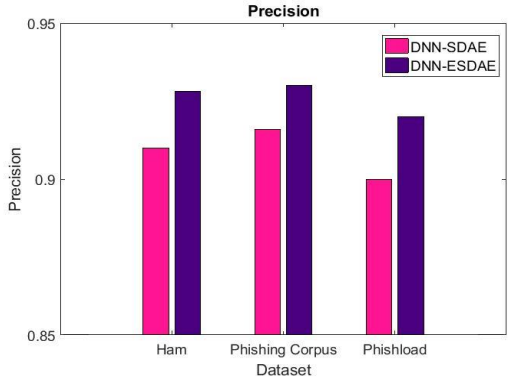

Figure 2. Evaluation of Precision

The precision of DNN-SDAE and DNN-ESDAE on three different datasets is shown in Figure 2. The precision of DNN-ESDAE is 1.98%, 1.53%, and 2.22% greater than DNN-SDAE on Ham, Phishing Corpus, and Phishload datasets respectively. From this analysis, it is known that the proposed DNN-ESDAE has high precision than DNN-SDAE for phishing attack detection.

C. Recall

It measures the completeness of the DNN results, i.e., what percentage of phishing URLs did the classifier label as phishing and it is calculated as,

$$Recall = \frac{TP}{TP + FN}$$

Table 3 shows the recall of DNN-SDAE and DNN-ESDAE on Ham, Phishing Corpus and Phishload datasets.

Table 3. Evaluation of Recall

Datasets	DNN-SDAE	DNN-ESDAE
Ham	0.91	0.919
Phishing Corpus	0.915	0.934
Phishload	0.92	0.94

The recall of DNN-SDAE and DNN-ESDAE on three different datasets is shown in Figure 3. The recall of DNN-ESDAE is 0.99%, 2.78%, and 2.17% greater than DNN-SDAE on Ham, Phishing Corpus, and Phishload datasets respectively. From this analysis, it is known that the proposed DNN-ESDAE has high recall than DNN-SDAE for phishing attack detection.

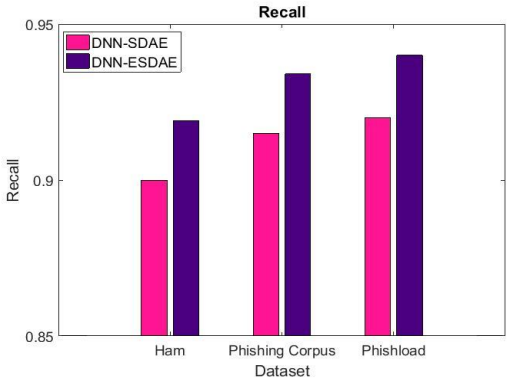

Figure 3. Evaluation of Recall

D. F-measure

It is the harmonic mean of precision and recall. It is calculated as,

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Table 4 shows the f-measure of DNN-SDAE and DNN-ESDAE on Ham, Phishing Corpus and Phishload datasets.

Table 3. Evaluation of F-measure

Datasets	DNN-SDAE	DNN-ESDAE
Ham	0.905	0.927
Phishing Corpus	0.915	0.93
Phishload	0.904	0.92

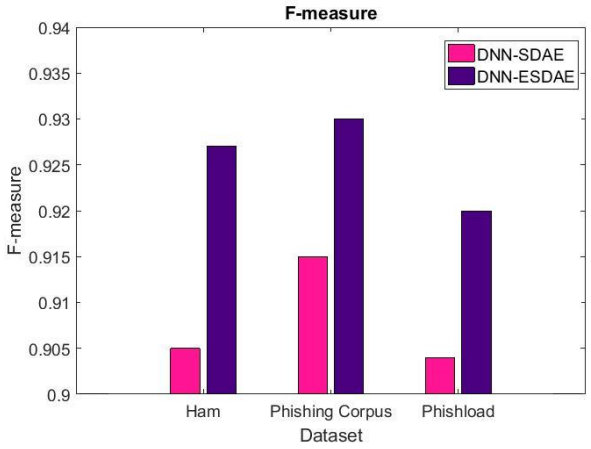

Figure 4. Evaluation of F-measure

The f-measure of DNN-SDAE and DNN-ESDAE on three different datasets is shown in Figure 4. The f-measure of DNN-ESDAE is 2.43%, 1.64%, and 1.77% greater than DNN-SDAE on Ham, Phishing Corpus, and Phishload datasets respectively. From this analysis, it is known that the proposed DNN-ESDAE has high f-measure than DNN-SDAE for phishing attack detection.

V. CONCLUSION AND FUTURE SCOPE

In this paper, DNN-ESDAE is proposed to enhance phishing attack detection based on the ensembling of SDAE. The extracted features of URL, HTML and domain are given as input to three different SDAE which reconstruct the input features. The negative correlation learning is introduced to fine-tune the softmax classifier in DNN. The best ensembling of SDAE is chosen by using SFLOA and majority voting is used to integrate the results of ESDAE. The experimental results prove that the proposed DNN-ESDAE has high accuracy, precision, recall and f-measure for Ham, Phishing Corpus and Phishload datasets than DNN-SDAE based phishing attack detection. Even though, DNN-ESDAE has better performance it has limitations such as non-uniform initial population, slow convergent rate, local searching ability, adaptive ability, and premature convergence because of using SFLOA. In the future, these limitations will overcome by developing a better phishing detection method.

REFERENCES

[1] K. Parsons, M. Butavicius, P. Delfabbro, M. Lillie, "*Predicting susceptibility to social influence in phishing emails*", International Journal of Human-Computer Studies, Vol.**128,** pp.**17-26, 2019**.

[2] M. Shukla, S. Sharma, "*Analysis of efficient classification algorithm for detection of phishing site",* International Journal of Scientific Research in Computer Sciences and Engineering, Vol.**5,** Issue.**3,** pp. **136-141, 2017**.

[3] T. Chin, K. Xiong, C. Hu, "*Phishlimiter: A Phishing Detection and Mitigation Approach Using Software-Defined Networking*", IEEE Access, Vol.**6,** pp.**42516-42531, 2018**.

[4] K. Sumathi, V. Sujatha, "*Deep learning based-phishing attack detection*", International Journal of Recent Technology and Engineering (IJRTE), Vol.**8,** Issue.**3,** pp.**8428-8432**, **2019**.

[5] H.K. Soni, "*Machine learning- A new paradigm of AI*", International Journal of Scientific Research in Network Security and Communication, Vol.**7,** Issue.**3,** pp.**31-32, 2019**.

[6] K. Sumathi, V. Sujatha, "*Deep neural network with stacked denoise auto encoder for phishing detection*", International Journal of Machine Learning and Networked Collaborative Engineering (IJMLNCE), Vol.**3,** Issue.**2,** pp.**114-124, 2019**.

[7] I.R.A. Hamid, J. Abawajy, "*Hybrid feature selection for phishing email detection*", In the Proceedings of 2011 International Conference on Algorithms and Architectures for Parallel Processing Springer, **Berlin**, pp.**266-275**, **2011**.

[8] G.A. Montazer, S. ArabYarmohammadi, "*Detection of phishing attacks in Iranian e-banking using a fuzzy–rough hybrid system*", Applied Soft Computing, Vol.**35,** pp.**482-492**, **2015**.

[9] G. Sonowal, K.S. Kuppusamy, "*PhiDMA–a phishing detection model with multi-filter approach*", Journal of King Saud University-Computer and Information Sciences, Vol.**32,** pp.**99-112, 2017**.

[10] S. Smadi, N. Aslam, L. Zhang, "*Detection of online phishing email using dynamic evolving neural network based on reinforcement learning",* Decision Support Systems, Vol.**107,** pp.**88-102, 2018**.

[11] P. Yang, G. Zhao, P. Zeng, "*Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning",* IEEE Access, Vol.**7,** pp.**15196-15209, 2019**.

[12] M. Chatterjee, A.S. Namin, "*Detecting Phishing Websites through Deep Reinforcement Learning*", In the proceedings of 2019 IEEE Conference on Annual Computer Software and Applications Conference (COMPSAC), Vol.**2,** pp.**227-232, 2019**.

[13] F. Toolan, J. Carthy, "*Feature selection for spam and phishing detection",* IEEE eCrime Researchers Summit, pp.**1-12, 2010**.

[14] S. Garera, N. Provos, M. Chew, A.D. Rubin, "*A framework for detection and measurement of phishing attacks*", In the Proceedings of the 2007 ACM workshop on Recurring malcode, pp.**1-8, 2007**.

[15] M. Lichman, UCI machine learning repository, **2013**.

Authors Profile

K. Sumathi is a student of CMS college of science and commerce, affiliated to Bharathiar university, Coimbatore, Tamilnadu, India. She is pursing Ph.D in Computer Science. She is doing research in the area of information security.

Dr. V. Sujatha has 16 years of teaching experience and 2 years of IT Industrial experience. Her area of specialization is web mining, IoT and Big Data Analysis. She has published 24 research articles in National and International Journals and also presented papers in several National Conferences, Seminars and Workshops. She is currently guiding M.Phil and Ph.D Scholars. She has an ideal knowledge in programming languages, DOT NET frameworks and has developed two live projects using Visual programming. She also sets question papers for universities in TamilNadu.

