International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 11 • Issue 2 pp. 1-8 April 2020 - Sept 2020 www.csjournals.com

Rough Deep Neural Network with Enhanced Ensembling Stacked De-noise Auto encoder for Phishing Attack Detection

Sumathi. K

ABSTRACT: Phishing is the process of stealing sensitive information such as social security number, username, password and credit card information etc with the help of fake webpage that imitates trusted website. Recent research uses Deep Neural Network (DNN) for phishing detection. The redundant and irrelevant features may affect the accuracy of DNN-based phishing detection. So, Stacked Denoise Auto Encoder (SDAE) was introduced to reconstruct the features for handling the redundant and irrelevant features. The reconstructed features by SDAE were processed in DNN for phishing detection. However, the computational complexity of DNN-SDAE was high since all features were processed in single DNN-SDAE. In order to reduce the computational complexity, DNN-Ensembling SDAE (DNN-ESDAE) was proposed in which different types of features were processed in different DNN-SDAE and the most optimal result of DNN-SDAE was obtained using Shuffled Frog Leaping Optimization Algorithm (SFLOA) and majority voting technique. The random initialization of frog population in SFLOA leads to some problems such as non-uniform initial population, slow searching speed in the late evolution and easily trapping into local maximum and minimum problem. In order to overcome this problem, an Adaptive SFLOA (ASFLOA) is proposed in this paper. In ASFLOA, the solution space is evenly split into certain number of parts and an individual is generated randomly in each part. It ensures the population be both uniform and random. In addition to this, DNN is combined with rough set theory to enhance the performance of DNN. The number of neurons in the hidden layer is determined by the size of the positive region and boundary region. Each neuron in the hidden layer has one lower approximation and upper approximation neuron. The lower and upper approximation is performed in the hidden layer neurons and their outputs are combined for phishing detection. Finally, the results of each RDNN-SDAE are ensemble and optimized by ASFLO and majority voting. The proposed whole process is named as Rough DNN-Enhanced ESDAE (RDNN-EESDAE).

Keywords: Phishing, Deep Neural Network, Adaptive Shuffled Frog Leaping Algorithm, Rough set theory, Rough Deep Neural Network.

1. INTRODUCTION

Phishing [1] is a form of online hazard that is described as the art of exemplifying an honest company website with a view to steal user's sensitive information such as social security numbers, usernames and passwords. Phishing websites are typically created to mimic legitimate websites by dishonest people. Such websites are extremely visual like legitimate ones to attempt and defraud trustworthy internet users. Mostly the phishing is performed through email where a dishonest people induce the receiver to click on a link or to open an attachment within an email. An efficient phishing email detection technique is more required to avoid this threat and to protect the user's personal or sensitive information.

Deep Neural Network (DNN) [2] was a deep learning technique was introduced for phishing email detection. Initially in DNN-based phishing email detection method, Uniform Resource Locator (URL), Hypertext Markup Language (HTML) and domain-based features were extracted using feature extractor. Then, the extracted features were processed in DNN for phishing email detection. The accuracy of DNN-based phishing email detection method was affected due to the irrelevant, redundant and noisy features. This problem was solved by using Stacked Denoise Auto Encoder (SDAE) [3] which reconstructs the URL, HTML and domain-based features and the reconstructed features were used in DNN for phishing email detection.

The computational complexity of DNN-SDAE based phishing email detection method was reduced by proposing DNN with Ensembling SDAE (DNN-ESDAE) [4] where three different features were separately processed in three SDAE. Then, the best phishing email detection results were selected by using Shuffled Frog Leaping Optimization Algorithm (SFLOA) and majority voting. However, the SFLOA has the problem of non-uniform initial population, slow searching speed in the late evolution and easily trapping into local maximum and minimum problem.

In order to solve these problems, mutation operator and population diversity are introduced in SFLOA for better optimal selection of phishing email detection results. It is named as Adaptive SFLOA (ASFLOA). In addition to this, the phishing detection accuracy is further enhanced by combining Rough set with DNN (RDNN). In RDNN, the number of neurons in hidden layer is determined by the sizes of the positive region and boundary

region which are defined rough set theory. It can reduce the blindness of selecting the number of neurons in the hidden layer. Then, the best results of RDNN are selected using ASFLOA and majority voting. It enhances the efficiency of phishing email detection.

2. LITERATURE SURVEY

Li et al. [5] proposed a semi-supervised learning approach for detection of phishing web pages. Initially, the features of web pages were extracted and it was integrated with color histogram, gray histogram and spatial relationship between sub graphs. After the feature extraction and integration, the features of sensitive information were examined by using page analysis based on Document Object Model (DOM). This information was given as input to the Transductive Support Vector Machine (TSVM) to train classifier for phishing web page detection. The learning rate of this approach will be further enhanced by analyzing the properties of web image and feature structure of the web page.

Montazer & ArabYarmohammadi [6] introduced a fuzzy-rough set hybrid system for phishing attack detection. It identified the influential features of phishing web sites and the most significant features were selected using rough set theory. The selected features were processed in fuzzy expert system for phishing detection. Nonetheless, membership function of fuzzy system has great impact on the efficiency of fuzzy-rough set hybrid system.

Moghimi et al. [7] proposed a rule-based phishing detection method. It used two novel feature sets including page resource identity and access protocol of page resource elements for finding the relationship between the URL and content of a page. It was calculated using approximate string matching algorithm. The proposed feature sets were combined with the subset of relevant features for creation of web page feature vector. It was given as input to Support Vector Machine (SVM) for phishing detection. However, proper selection of kernel function in SVM is more difficult.

Tan et al. [8] proposed a phishing detection technique based on the difference between the actual and target identities of a webpage. Initially, identity keywords were extracted from the textual content of the webpage using N-gram model. Then, a search engine was used to find the target domain name based on the identity-relevant features. Finally, 3-tier identity matching system was applied on the determined target domain to find the legitimacy of the query webpage. However, this technique will be enhanced with an optical character recognition to address visual cloning problems.

Şahingöz et al. [9] proposed a machine learning based phishing detection method. Initially, different features such as hybrid features, Natural Language Processing (NLP)-based features and word vectors were extracted. The extracted features were processed in Random Forest (RF), decision tree, Naïve Bayes (NB), Sequential Minimal Optimization (SMO), k-Nearest Neighbor (kNN) and k-star classifiers to classify the URLs as legitimate URL or phishing URLs. However, it is not more preferred for real-time detection.

Li et al. [10] proposed a real time phishing webpage detection system to protect the user's sensitive information. In this system, the features from URLs and HTMLs codes were extracted and combined them as feature vectors. This feature vectors were given as input to the stacking model which is the combination of multiple machine learning models such as XGBoost, LightGBM and Gradient Boosting Decision Tree (GBDT) for phishing webpage detection. However, this system has high time consumption problem because it needs multi-page information for phishing webpage detection.

Orunsolu et al. [11] proposed a predictive model based on machine learning techniques for phishing detection. A feature selection module based on incremental-component based system was used in the predictive model to extract the features from webpage behavior, URL and webpage properties. The extracted features were given as input to Support Vector Machine (SVM) and Naïve Bayes (NB) for identification of phishing contents in online communication systems. However, the efficiency of this model depends on the proper selection of kernel function and amount of data used in Naïve Bayes (NB).

3. PROPOSED METHODOLOGY

In this section, the proposed RDNN-EESDAE for phishing email detection is described in detail. A feature extractor gets URL and web based code as input and returns URL, HTML and domain-based features [12-14]. Three different types of features are processed in three different SDAE which reconstructs the input features. After the training process of SDAE, the weight and bias values of the encoder layer in SDAE are taken as an initialization of a DNN's hidden layer. The number of hidden layers used in DNN is determined based on the rough set theory. The best ensembling of DNN-SDAE is determined by using ASFLOA and majority voting.

3.1 RDNN-ESDAE FOR PHISHING EMAIL DETECTION

Structure of RDNN

The reconstructed features from SDAE and its trained weight and bias values are initialized in the hidden layer of RDNN. The RDNN is the combination of rough set theory and DNN. The neurons in the hidden layer of DNN are rough neurons which are trained by reconstructed features divided by rough set. Rough set theory

divides the whole reconstructed features into two unique parts as upper approximation and lower approximation those are trained by upper approximation set and lower approximation set correspondingly. The input weight and biases of upper and lower approximation neurons are obtained from the SDAE. The output weights of upper and lower approximation neurons are determined as the method of DNN. Even though the training process of upper and lower approximation neurons is relatively independent, the phishing email detection results of RDNN is decided by the outputs of upper and lower approximation neurons. RDNN closely integrates DNN with rough set theory and is used for guiding the learning process of RDNN to split the reconstructed features by rough set.

RDNN for phishing email classification

Each neuron in the hidden layer of RDNN contains one lower approximation and upper approximation neuron. For a reconstructed feature set Fea, A is the condition attributes set and D is the decision attribute set. So, Fea is divided into Fea_{low} (lower approximation set) Fea_{up} (upper approximation set) which are given as follows:

$$Fea_{low} = POS_B(D) = \bigcup_{Y \in Fea/D} \underline{R_B} Y \quad (3.1)$$

$$Fea_{up} = \overline{R_B} Fea \quad (3.2)$$

In Eq. (3.1), $POS_B(D)$ is the B-positive region of D. Assume H is the number of neurons in the hidden layer in RDNN, w_{low} is the input weights connecting with lower approximation neurons and b_{low} is the biases of lower approximation neurons. The lower approximation neurons of EDNN are trained by Fea_{low}. According to DNN, if $H \leq Num_H$, the hidden layers are defined as softmax activation function which is given as follows:

$$\sigma(Fea)_j = \frac{e^{Fea_j}}{\sum_{h=1}^{H} e^{Fea_k}} \quad (3.3)$$

In Eq. (3.3), $j = 1,2, \ldots H$. Each feature in the reconstructed feature has its own weight values $w_1, w_2, \ldots w_H$ and the weighted sum of the reconstructed features is done by the adder function as follows:

$$u = \sum_{low=1}^{H} w_{low}\, Fea_{low} \quad (3.4)$$

The output of lower approximation neurons are given as follows:

$$Output_{low} = \sigma\left(\sum_{low=1}^{H} w_{low} Fea_{low} + b_{low}\right) \quad (3.5)$$

If $H > Num_H$, the hidden layers are defined as softmax activation function which is given as follows:

$$\sigma(Fea)_j = \frac{e^{Fea_j}}{\sum_{h=1}^{H} e^{Fea_k}} \quad (3.6)$$

In Eq. (3.6), $j = 1,2, \ldots H$. Each feature in the reconstructed feature has its own weight values $w_1, w_2, \ldots w_H$ and the weighted sum of the reconstructed features is done by the adder function as follows:

$$u = \sum_{up=1}^{H} w_{up}\, Fea_{up} \quad (3.7)$$

The output of lower approximation neurons are given as follows:

$$Output_{up} = \sigma\left(\sum_{low,up=1}^{H} w_{up} Fea_{low} + b_{up}\right) \quad (3.8)$$

The final output of RDNN is given as follows:

$$Output = \max\left(Output_{low}, Output_{up}\right) \times \min\left(Output_{low}, Output_{up}\right) \quad (3.9)$$

Removal of redundant features

The redundant features may affect the phishing email classification results. So the removal of redundant features is introduced in RDNN-ESDAE using rough set theory. For the extracted URL, HTML and domain-based features X, C is the condition attribute set, D is the decision attribute set and $B \subseteq C; \forall a \in C - B$, the importance of the feature a for the decision attribute set D based on the condition attributes set B is as:

$$imp(a, B, D) = \gamma_{B \cup \{a\}}(D) - \gamma_B(D) \quad (3.10)$$

In Eq (3.10), $\gamma_B(D)$ is the approximate quality of B for D which is calculated as,

$$\gamma_B(D) = \frac{|POS_B(D)|}{|Fea|} = \frac{\left|\bigcup_{x \in Fea/D} \underline{R_B} x\right|}{|Fea|} \quad (3.11)$$

The greater the value of $imp(a, B, D)$ is the more significant feature a is for D. Based on the value of $imp(a, B, D)$, removes the non-redundant features.

3.2 RDNN-EESDAE FOR ENSEMBLING AND DETECTION OF BEST PHISHING EMAIL DETECTION RESULT

The results of RDNN-ESDAE are ensembling and the best result is chosen using ASFLOA. The random initialization of frog population leads to non uniform initialization population and get into the local maximum and minimum problem. In order to overcome this problem, the solution space is evenly split into p parts. Then, in each part of the solution space, an individual is generated randomly. This ensures the population be both random and uniform. It can increase the efficiency of phishing email detection and helps to escape from local extremum problem. After the initialization of population, it is split into searching population and competing population. In the searching population, the optimal solution (best ensembling results) is saved and the second best solution is updated. In the competing population, the mutation operator is imported to add the genotype.

Searching population

During the searching evolution process, the second best solution is added in the RDNN-EESDAE. The optimal solution easily traps into the local optimum along with the evolution. So, developing a new possible solution is important. After the evolution process, both the best solution and the second best solutions are updated in ASFLOA. The second best solution is changed by the simplex method. By updating the second best solution, the search space is reduced in the evolution process. Also, it enhances the local searching ability of SFLOA.

Competing Population

The original population includes a mutation operator and acts on individuals that gave a low convergence rate. It enhances the convergence speed of the whole population. The population objective function is defined to represent the convergence degree of the population. The convergence degree is given as follows:

$$\delta^2 = \frac{1}{F(\mathcal{X})}\sum_{j=1}^{n}\left(F(\mathcal{X})_j - F(\mathcal{X})_{avg}\right)^2 \quad (3.12)$$

In Eq. (3.12), $F(\mathcal{X})_j$ is the fitness value of the individual, $F(\mathcal{X})_{avg}$ is the average fitness of the population and N is the quantity of the population. $F(\mathcal{X})$ is given as follows:

$$F(\mathcal{X}) = \frac{1}{N}\sum_{n=1}^{N}\left(Fea_i(n) - y(n)\right)^2 - E\left[\left(\frac{x-\mu}{\sigma}\right)^4\right] \quad (3.13)$$

The smaller the δ^2 is the stronger the convergence is. To change the searching space, consider

$$\omega_n[i] = low_w + (high_w - low_w) \times S[i] \quad (3.14)$$

In Eq. (3.14), $\omega_n[i]$ is the adaptive compressibility factor of the frog in n generation, $high_w$ and low_w are the biggest and smallest compressibility factor respectively.

$$S[i] = \frac{F(\mathcal{X})\_worst_n - F(\mathcal{X})_{n,i}}{F(\mathcal{X})\_worst_n - F(\mathcal{X})\_best_n} \quad (3.15)$$

In Eq. (3.15), $F(\mathcal{X})_{n,i}$ is the fitness value of the ith frog in n generation, $F(\mathcal{X})\_worst_n$ and $F(\mathcal{X})\_best_n$ are the biggest and smallest of the n generation. Based on the fitness value, the compressibility factor of each frog is also changed. The moving step of frog i is changed as,

$$D_i = rand() \times (\mathcal{X}_{best} - \mathcal{X}_{worst}) \times \omega \quad (3.16)$$

In Eq. (3.16), $rand()$ is the random number ranges from 0 to 1, $\mathcal{X}_{best}$ best solution obtained from RDNN-ESDAE and $\mathcal{X}_{worst}$ worst solution obtained from RDNN-ESDAE. The worst frog in each population is updated as,

$$\mathcal{X}_{w,1}^{new} = \mathcal{X}_{w,0} + D_i \quad (3.17)$$

If the evolution generates a better frog, it replaces the worst frog otherwise $\mathcal{X}_{best}$ is replaced by global fitness and the process is continued. If the objective function of the new frog is not better than objective function of $\mathcal{X}_{worst}$, 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. Hence, the local search in each sub-memeplex is completed and the sub-memplexes are returned to memeplexes. After that, shuffling process is initialized and a new generation is created. The leaping and shuffling process is continued until the convergence is satisfied. Hence, the optimal ensembling of RDNN-EESDAE is obtained based on ASFLOA. At last, majority voting is conducted on the output of the final ensembling to detect the phishing email effectively.

4. RESULT AND DISCUSSION

In this section, the performance of DNN-ESDAE and RDNN-EESDAE is tested in terms of accuracy, precision, recall and f-measure. Ham, Phishing Corpus and Phishload datasets [3] are three different datasets are used for the experimental purpose.

4.1 Accuracy

Accuracy calculates the overall rate of correctly detected phishing and legitimate URLs. It is calculated as,

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

where, if the class label is phishing URL and the phishing detection outcome is phishing URL, then it is TP
If the class label is legitimate URL and the phishing detection outcome is legitimate URL, then it is TN
If the class label is legitimate URL and the phishing detection outcome is phishing URL, then it is FP
If the class label is phishing URL and the phishing detection outcome is legitimate URL, then it is FN
The accuracy value of DNN-ESDAE and RDNN-EESDAE on three different datasets is tabulated in Table 1.

Table.1 Evaluation of Accuracy

Datasets	DNN-ESDAE	RDNN-EESDAE
Ham	0.94	0.97
Phishing Corpus	0.957	0.98
Phishload	0.939	0.96

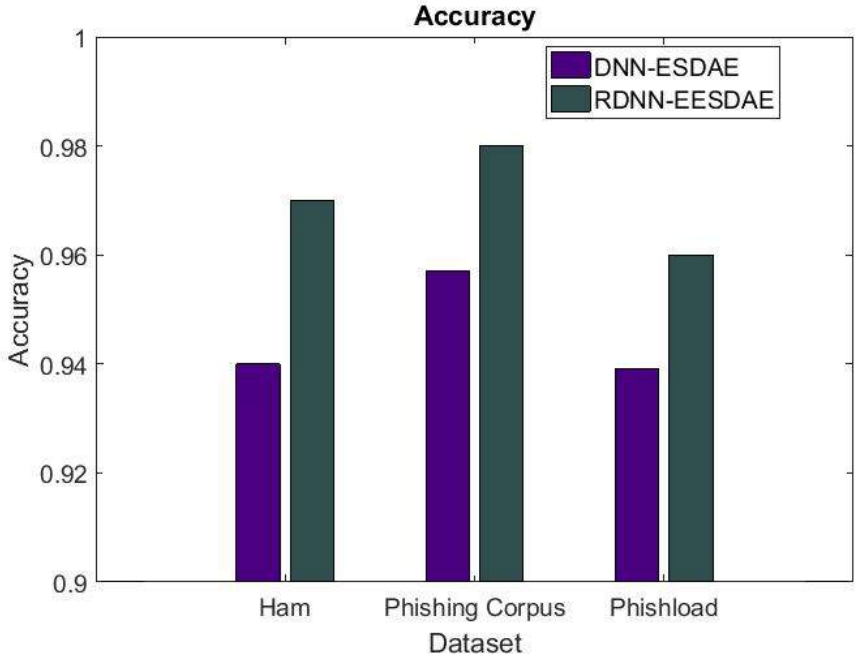

Figure.1 Evaluation of Accuracy

Figure 1, shows the accuracy of DNN-ESDAE and RDNN-EESDAE on Ham, Phishing Corpus and Phishload datasets. In Ham dataset, the phishing detection accuracy of RDNN-EESDAE is 3.19% greater than DNN-ESDAE. From this analysis, it is proved that the RDNN-EESDAE has high accuracy than DNN-ESDAE.

4.2 Precision

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

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

The precision value of DNN-ESDAE and RDNN-EESDAE on three different datasets is tabulated in Table 2.

Table.2 Evaluation of Precision

Datasets	DNN-ESDAE	RDNN-EESDAE
Ham	0.928	0.954
Phishing Corpus	0.93	0.963
Phishload	0.92	0.957

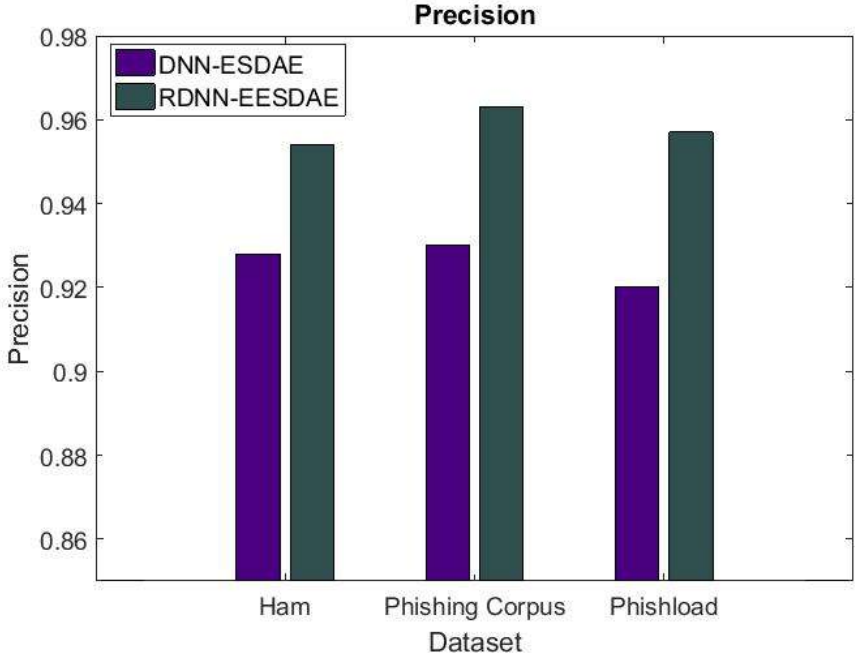

Figure.2 Evaluation of Precision

Figure 2, shows the precision of DNN-ESDAE and RDNN-EESDAE on Ham, Phishing Corpus and Phishload datasets. In Ham dataset, the phishing detection precision of RDNN-EESDAE is 2.8% greater than DNN-ESDAE. From this analysis, it is proved that the RDNN-EESDAE has high precision than DNN-ESDAE.

4.3 Recall

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

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

The recall value of DNN-ESDAE and RDNN-EESDAE on three different datasets is tabulated in Table 3.

Table.3 Evaluation of Recall

Datasets	DNN-ESDAE	RDNN-EESDAE
Ham	0.919	0.934
Phishing Corpus	0.934	0.956
Phishload	0.94	0.967

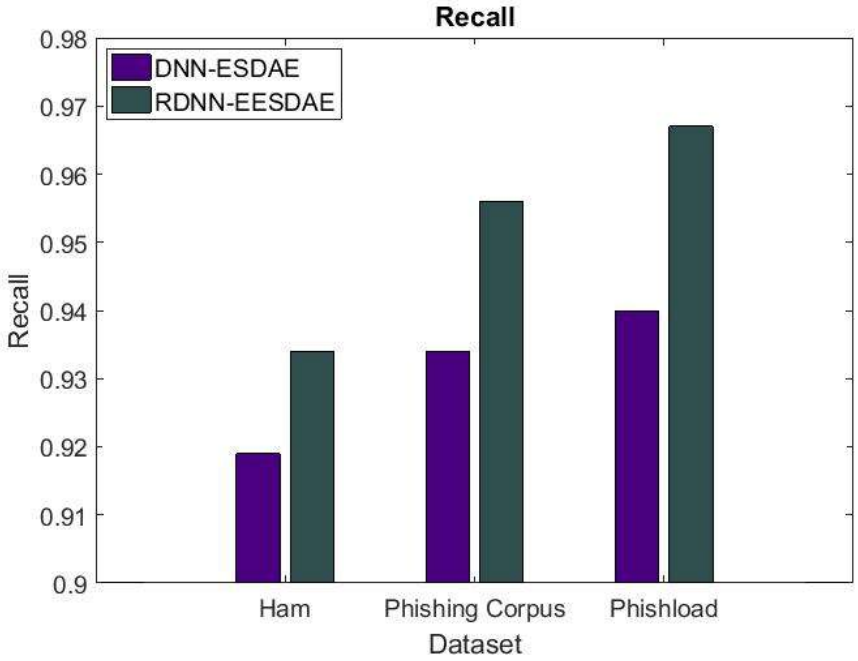

Figure.3 Evaluation of Recall

Figure 3, shows the recall of DNN-ESDAE and RDNN-EESDAE on Ham, Phishing Corpus and Phishload datasets. In Ham dataset, the phishing detection precision of RDNN-EESDAE is 1.63% greater than DNN-ESDAE. From this analysis, it is proved that the RDNN-EESDAE has high recall than DNN-ESDAE.

4.4 F-measure

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

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

The f-measure value of DNN-ESDAE and RDNN-EESDAE on three different datasets is tabulated in Table 4.

Table.4 Evaluation of F-measure

Datasets	DNN-ESDAE	RDNN-EESDAE
Ham	0.927	0.954
Phishing Corpus	0.93	0.963
Phishload	0.92	0.951

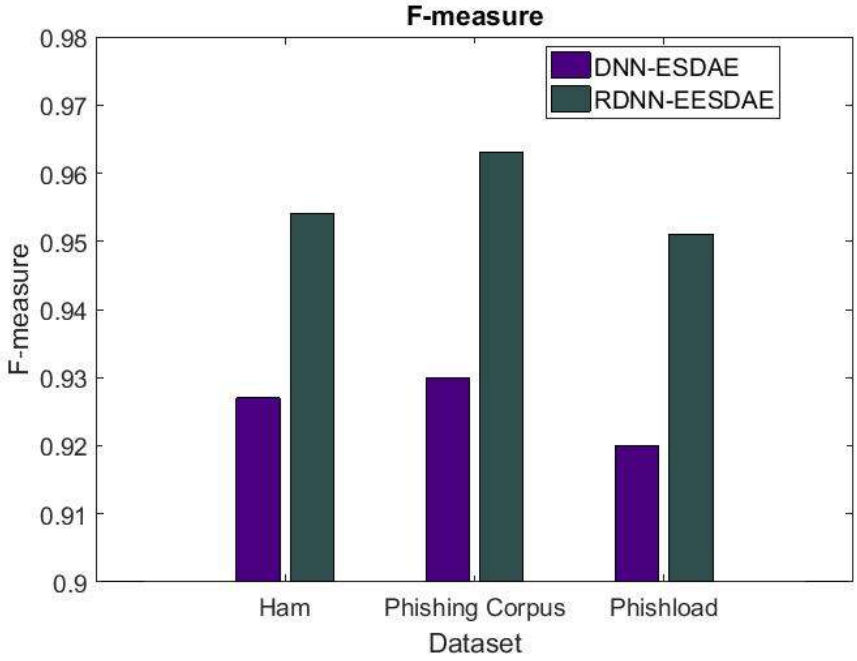

Figure.4 Evaluation of F-measure

Figure 4, shows the f-measure of DNN-ESDAE and RDNN-EESDAE on Ham, Phishing Corpus and Phishload datasets. In Ham dataset, the phishing detection precision of RDNN-EESDAE is 2.91% greater than DNN-ESDAE. From this analysis, it is proved that the RDNN-EESDAE has high recall than DNN-ESDAE

5. CONCLUSION

In this paper, RDNN-EESDAE is proposed for phishing detection. Initially, a feature extractor extracts the URL, HTML and domain based features from the URL. Then, these features are reconstructed using SDAE. After the training process, the weight, bias values and the reconstructed features are given as input to DNN. The number of neurons in the hidden layers is determined by the concept of rough set theory which reduces the computational complexity for phishing detection. In RDNN, each neuron in the hidden layer has lower and upper approximation. The weight and bias values of lower and upper approximation neurons are obtained from SDAE and DNN respectively. Finally, the outputs of lower and upper approximations are combined for phishing detection. The best ensembling of RDNN-ESDAE is obtained using ASFLOA and majority voting. The AFSLOA introduces mutation operator and population diversity in SFLOA to solve the non-uniform initial population, slow searching speed in the late evolution and easily trapping into local maximum and minimum problems. The experimental results prove that the proposed RDNN-EESDAE has high accuracy, precision, recall and f-measure than DNN-ESDAE for Ham, Phishing Corpus and Phishload datasets.

REFERENCES

[1] Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection based associative classification data mining. *Expert Systems with Applications*, *41*(13), 5948-5959.

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

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

[4] Sumathi, K., & Sujatha, V. (2019). Ensembling of stacked denoise autoencoder for phishing attack detection. *International Journal of Computer Sciences and Engineering*, *7*(12), 115-121.

[5] Li, Y., Xiao, R., Feng, J., & Zhao, L. (2013). A semi-supervised learning approach for detection of phishing webpages. *Optik*, *124*(23), 6027-6033.

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

[7] Moghimi, M., & Varjani, A. Y. (2016). New rule-based phishing detection method. *Expert systems with applications*, *53*, 231-242.

[8] Tan, C. L., Chiew, K. L., & Wong, K. (2016). PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. *Decision Support Systems*, *88*, 18-27.

[9] Şahingöz, Ö. K., Buber, E., Demir, Ö., & Diri, B. (2017). Machine Learning Based Phishing Detection from URIs. *Expert System with Applications*, *117*(2019), 345-357.

[10] Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2019). A stacking model using URL and HTML features for phishing webpage detection. *Future Generation Computer Systems*, *94*, 27-39.
[11] Orunsolu, A. A., Sodiya, A. S., & Akinwale, A. T. (2019). A predictive model for phishing detection. *Journal of King Saud University- Computer and Information Sciences.*
[12] Toolan, F., & Carthy, J. (2010, October). Feature selection for spam and phishing detection. In *2010 eCrime Researchers Summit* (pp. 1-12). IEEE.
[13] Garera, S., Provos, N., Chew, M., & Rubin, A. D. (2007, November). A framework for detection and measurement of phishing attacks. In *Proceedings of the 2007 ACM workshop on Recurring malcode* (pp. 1-8). ACM.
[14] Lichman, M. (2013). UCI machine learning repository.