Vol. 03, No. 2,(2019),114–124, **ISSN:2581-3242**

IJMLNCE
JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2019v03i02.005

Deep Neural Network with Stacked Denoise Auto Encoder for Phishing Detection

[a]Kothandan Sumathi[*], [b]Vijayan Sujatha

[a]*Research scholar, Department of Computer Applications, CMS College of Science and Commerce, Coimbatore, Tamilnadu, India*

[b]*Associate Professor, Department of Computer Applications, CMS College of Science and Commerce, Coimbatore, Tamilnadu, India*

[a]*nksmba@rediffmail.com, (https://orcid.org/0000-0001-8761-6905)*

[b]*sujathapadmakumar4@gmail.com, (https://orcid.org/0000-0002-5473-0362)*

Abstract

Sensitive information such as credit card information, username, password and social security number etc, can be stolen using a fake page that imitates trusted website is called phishing. The attacker designs a similar webpage either by copying or making small manipulation to the legitimate page so that the online user cannot distinguish the legitimate and fake websites. A Deep Neural Network (DNN) was introduced to detect the phishing Uniform Resource Locator (URL). Initially, a 30-dimension feature vector was constructed based on URL-based features, Hypertext Markup Language (HTML)-based features and domain-based features. These features were processed in DNN to detect the phishing URL. However, the irrelevant, redundant and noisy features in the dataset increase the complexity of DNN classifier. So the feature selection is required for efficient phishing attack detection. But feature selection is a time-consuming process since it is an independent process. So in this paper, a feature vector is generated by DNN itself using Stacked Denoise Auto Encoder (SDAE). Moreover, the noisy data such as missing features affect the efficiency of phishing detection so the SDAE is trained to reconstruct a clean input feature vector. The initial input feature vector is corrupted by setting some feature vectors as zero. Then, the corrupted feature vector is then plotted with basic auto encoder, to a hidden representation from that the input feature vector is reconstructed. The reconstructed features are given as input to DNN which selects the most relevant features and predicts the phishing URL. Hence the sparse feature representation of SDAE increases the classification accuracy of DNN. The experiments are conducted in Ham, Phishing Corpus and Phishload datasets to prove the effectiveness of DNN-SDAE.

Keywords

Phishing

Deep Neural Network

Auto encoder

Stacked Denoise

[*] Corresponding author

KothandanSumathi

Email: nksmba@rediffmail.com

1. Introduction

Phishing [1] is a treacherous attempt wherein the main intention of the attacker is to capture a victim's sensitive information. Generally a classic phishing attack is performed based on using a proxy or man-in-the-middle attacks or making use of browser weakness. However, the most general method is creating a web page which is more similar to the one which is familiar to the user. Hence, phishing still causes an important security threat and a large number of internet users faces this problem. Such attacks are also causing trouble for companies that provide online services.

Various techniques such as Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes and Artificial Neural Network (ANN) [2] were utilized for phishing attack detection. Good quality of training data is obtained by using deep learning technique. So, a Deep Neural Network (DNN) [3] was introduced for phishing attack detection. Initially, a feature extractor was used to extract 30-dimension feature vector from the URLs. Then, the extracted features were processed in DNN to classify the URL as a phishing URL or legitimate URL. The dataset may consist of redundant, irrelevant and noisy features. By using such features in DNN, the classification accuracy of DNN is affected.

So in this paper, DNN with Stacked Denoise Auto Encoder (SDAE) [4, 5] is proposed to remove the irrelevant, redundant and noisy data (missing features) for phishing URL detection. The feature selection process can remove irrelevant, redundant and noisy (missing features) features by selecting the most important features. But feature selection is a time-consuming process since it is an independent process. So, the feature vector is generated by DNN through SDAE. SDAE reconstructs the features from the corrupted version of the features by using a stochastic mapping. The reconstructed features are fed into the DNN to choose the most relevant features and detect the phishing URLs. The SDAE constructs high-level features which increase the classification accuracy of DNN.

2. Literature Survey

Hybrid feature selection method [6] was proposed for phishing email detection. This method was based on the grouping of behavior-based and content-based phishing detection approaches. Based on the email header, this method mined the behavior of the attacker. By analyzing the attacker behavior, it came to know that phishing email which had tended to generated from more than one domain. It indicated abnormal activity. However, this model concentrated only on email headers.

A novel approach was proposed [7] to detect and prevent from phishing URL. It combined Webpage similarity and URL-based based detection methods. URL-based phishing detection involved the extraction of actual URL and the result generated by the approach proceeded to the next phase. The approach proceeded to the visual similarity-based detection when the URL-based detection doesn't detect phishing. The webpage similarity-based detection used a threshold value for phishing attack detection. However, the efficiency of this approach depends on the threshold value.

A dynamic evolving neural network [8] model was proposed based on reinforcement learning for the detection of an online phishing email. It detected phishing emails by combined reinforcement learning and neural network as a single framework. It could adjust itself to produce email detection system. The dynamic model accepted the concept of reinforcement learning which dynamically enhanced the performance of the dynamic model. By including additional dataset to the offline dataset, the richness of this model could be increased.

An ensemble model [9] was proposed for detection of phishing attacks with Remove-Replace Feature Selection Technique (RRFST). It reduced the number of features by randomly selecting a feature and removing that feature when the attack detection accuracy was unchanged. On other hand, features were replaced to its original feature space when attack detection accuracy was increased. The selected features were used in Random Forest (RF), C4.5 and Classification and Regression Tree (CART). Even though the ensemble method had high detection accuracy, it consumed more space to store trees.

An efficient phishing website detection model [10] was proposed based on improved Back Propagation (BP) Neural Network (BPNN) and dual feature evaluation. A grey wolf algorithm was

introduced to fine tune the BPNN and dual feature to sensibly choose starting parameters. After that, a dual feature evaluation mechanism was utilized to estimate the results of improved BPNN. The phishing website recognition accuracy was improved by using the dual feature evaluation mechanism. By using more features, the accuracy of this model could be increased.

A fuzzy rough set feature selection method [11] was introduced to enhance phishing attack detection. Fuzzy Rough Set theory was used to select the most discriminative features in the dataset. The selected features were fed into RF, multipreceptron and Sequential Minimal Optimization (SMO) classifiers to find the phishing websites. However, the accuracy of this method is low.

Hybrid Ensemble Feature Selection (HEFS) method [12] was proposed for phishing detection. Initially, primary feature subsets were generated by using a Cumulative Distribution Function gradient (CDF-g) algorithm and those features were acted as input to the data perturbation ensemble method. It produced secondary feature subsets. Then, a group of features primary and secondary features were obtained by employing a function perturbation ensemble method. These features were processed by Random Forest (RF) to distinguish the phishing and legitimate websites. However, the complexity of RF is high due to the creation of more number of trees.

A Case-Based Reasoning Phishing Detection System (CBR-PDS) [13] was introduced to detect the phishing websites. It primarily based on the CBR which act as animportant part of phishing detection system. This system was highly flexible and active as it can easily detected latest phishing attacks. The CBR classifier classified websites with a relatively small dataset but other classifiers required to be trained in advance before classifying the websites. Initially CBR-PDS process checks OPT of current URL and checks whether the OPT was exist or not. If the OPT was present, the proposed CBR-PDS flag it as phishing website otherwise extracted features of that URL and it was formulated a new case to be tested. Then it starts CBR process which retrieves the most similar cases. However, it was failed to implement in integrated web-based CBR-PDS system.

3. Proposed Methodology

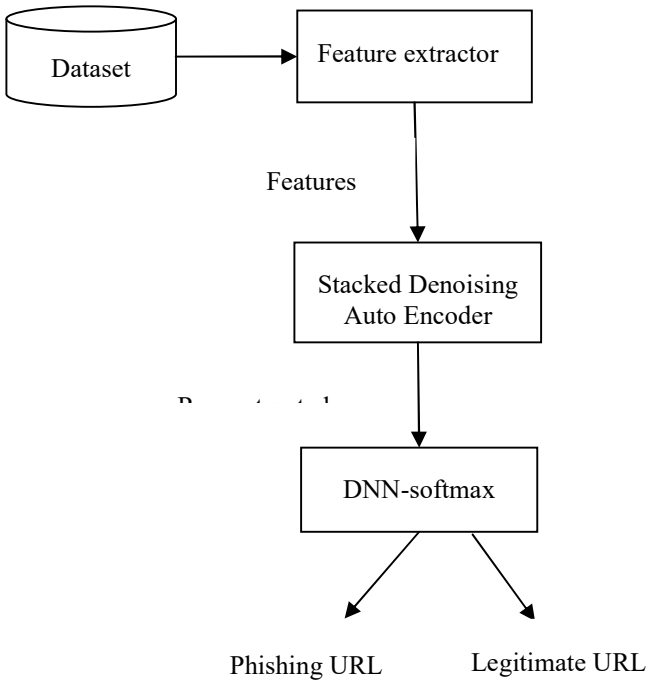

Fig.1: Workflow of Proposed Methodology

Here, the proposed Deep Neural Network with Stacked Denoising Auto Encoder (DNN-SDAE) for phishing attack detection is described in detail. Initially, a feature extractor [3] obtains URLs as an input and returns a vector that consists of thirty features. Then, the extracted features are reconstructed by SDAE to obtain high-level features for phishing detection. The reconstructed features are used in DNN which selects most significant features and predict the phishing URL. Fig. 1 shows the work flow of the proposed methodology.

3.1. Reconstruction of features using Stacked Denoise Auto Encoder

Stacked Denoising Auto Encoder (SDAE) gets the extracted features as input. SDAE has three layers are input layer, hidden layer and output layer. The hidden layer and output layer of SDAE is called as encoder layer and decoder layer respectively. SDAE consists of two encoding and two decoding layers. Initially, SDAE generates a vector $\tilde{x}$ by assigning some feature vectors as 0. Then, the SDAE uses $\tilde{x}$ as input feature vector. The number of units in the input layer is equal to the dimension of $\tilde{x}$ and the dimension of $\tilde{x}$ is denoted as d. In the encoding layer, the result of the first encoding layer acts as input to the second encoding layer. Suppose, there are L hidden layers in the encoding part, there is the activation function of the kth encoding layer is,

$$y^{(k+1)} = f_e\left(W^{(k+1)}y^{(k)} + b^{(k+1)}\right), k = 0,1,\dots,L-1 \quad (1)$$

In (1), the input y^0 is the original data x, $W^{(k+1)}$ is the input-to-hidden weights of the $k+1$th encoding layer, $b^{(k+1)}$ denotes the bias of the $k+1$th encoding layer and $f_e()$ is the activation function of the hidden layer. $f_e()$is formulated as,

$$f_e\left(W^{(k+1)}y^{(k)} + b^{(k+1)}\right) = max\left(0, W^{(k+1)}y^{(k)} + b^{(k+1)}\right) \quad (2)$$

If $W^{(k+1)}y^{(k)} + b^{(k+1)} < 0$, the output of the hidden layer will be zero. Hence it generates a sparse feature representation which may have better partition ability. The result of the last encoding layer is the high-level features extorted by SDAE. In the decoding part, the result of first decoding layer acts as input to the second decoding layer. The decoding function of the kth decode layer is given as follows:

$$z^{(k+1)} = f_d\left(W^{(L-k)T}z^{(k)} + b'(k+1)\right), k = 0,1,\dots L-1 \quad (3)$$

In (3), the inputof the first decoding layer is represented as $z^{(0)}$and the output of the last encoding layer is represented as $y^{(L)}$. Here, softplus function is used as decoding function $f_d()$ which is given as follows,

$$f_d(a) = \begin{cases} \log(1+e^a), & x \in [0,1] \\ a, & otherwise \end{cases} \quad (4)$$

In (4),$a = W^{(L-k)T}z^{(k)} + b'(k+1)$. The output $z^{(L)}$ of the last decoding layer is the reconstruction of the original input feature x. The objective function (reconstruction error) is given as follows,

$$Obj = \begin{cases} -\frac{1}{k}\sum_{i=0}^{k}\sum_{j=1}^{d}\left[x_j^{(i)}\log\left(z_j^{(i)}\right) + \left(1-x_j^{(i)}\right)log\left(1-z_j^{(i)}\right)\right] + \frac{\lambda}{2}\|W\|^2, & x \in [0,1] \\ \frac{1}{k}\sum_{i=1}^{k}\left\|x^{(i)} - z^{(i)}\right\|^2\frac{\lambda}{2}\|W\|^2, & otherwise \end{cases} \quad (5)$$

The process of SDAE while training is given as follows,

1. Get the features from the feature extractor

2. Train the initial encoding and the final decoding layer.

3.Getthe $W^{(l)}$, $b^{(l)}$and the features $y^{(l)}$ which are the output of the initial encoding layer.

4. Exploit $y^{(k)}$as the input data of the $(k+1)$th encoding layer.

5. Train the $(k+1)$th denoise auto encoder and get$W^{(k+1)}$ and $b^{(k+1)}$ and the features $y^{(k+1)}$.

The SDAE is trained independently and the features, weight and bias values in the last decoding layer acted as the inputs of the DNN.

3.2. Deep Neural Network with Softmax

DNN with softmax is used to minimize the objective function Obj. Given an input training set with n features $\{x^{(i)}\}_{i=1}^{n}$ which are selected by DNN and the feature's label set $\{t^{(i)}\}_{i=1}^{n}$, where $t^{(i)}$ is either -1 or 1, -1 represents the legitimate URL and 1 represents the phishing URL. Softmax estimates the probability of each feature vector belonging to each class (legitimate or phishing URL). The probability is given by,

$$P(t^{(i)}=2|x^{(i)};\theta)=\frac{1}{e^{\theta_1^T x^{(i)}}+e^{\theta_2^T x^{(i)}}}\begin{bmatrix}e^{\theta_1^T x^{(i)}}\\ e^{\theta_2^T x^{(i)}}\end{bmatrix} (6)$$

In (6), θ is the parameter of softmax parameter, $\frac{1}{e^{\theta_1^T x^{(i)}}+e^{\theta_2^T x^{(i)}}}\begin{bmatrix}e^{\theta_1^T x^{(i)}}\\ e^{\theta_2^T x^{(i)}}\end{bmatrix}$ normalizes the distribution so that the summation of the probability is one. The objective function of softmax is given as follows:

$$Obj(\theta)=-\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{h}1\{t^{(i)}=j\}\log\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{h}e^{\theta_h^T x^{(i)}}+e^{\theta_1^T x^{(i)}}}\right] \quad (7)$$

In (7),$h=1,2,$ $1\{.\}$ is the indicator function which returns 1 if the condition is true. Otherwise, it returns 0. By minimizing the objective function, legitimate URLs are detected effectively. The softmax is used to generate the classes of the features. The DNN together with SDAE and softmax is achieved an improved classification result.

4. Result and Discussion

The efficiency of DNN and DNN-SDAE is evaluated in this section based on accuracy, precision, recall and f-measure performance metrics. The DNN and DNN-SDAE based phishing detection in MATLAB 2018a by using Ham, Phishing Corpus and Phishload datasets. The Ham dataset is utilized for a baseline evaluation which consists of 4,150 legitimate e-mail communication and 1897 spam-based e-mails. The Phishing Corpus is utilized for its complication of phishing e-mail communication which consists of 4,559 phishing e-mail messages. The Phishload dataset is utilized for its raw web-based coding structures that consist of 1,185 legitimate and 3,718 phishing URLs.

4.1. Accuracy

Accuracy is the fraction of the total number of correct phishing URL detections to the actual dataset size. It measures the overall rate of correctly detected legitimate and phishing URLs.

$$Accuracy=\frac{True\ Positive+True\ Negative}{True\ Positive+True\ Negative+Fal\quad Positive+Fals\quad Negative} (8)$$

In (8), True Positive is the percentage of phishing URLs in the training dataset that is properly differentiated as phishing URLs

True Negative is the percentage of legitimate URLs in the training dataset that is properly differentiated as legitimate URLs

False Positive is the percentage of legitimate URLs that is improperly differentiated as phishing URLs

False Negative is the percentage of phishing URLs that is improperly differentiated as legitimate URLs.

Table 1 shows the comparison between DNN and DNN-SDAE in terms of accuracy for different datasets.

Table 1: Comparison of Accuracy

Datasets	**DNN**	**DNN-SDAE**
Ham	0.9	0.92
Phishing Corpus	0.92	0.94
Phishload	0.89	0.914

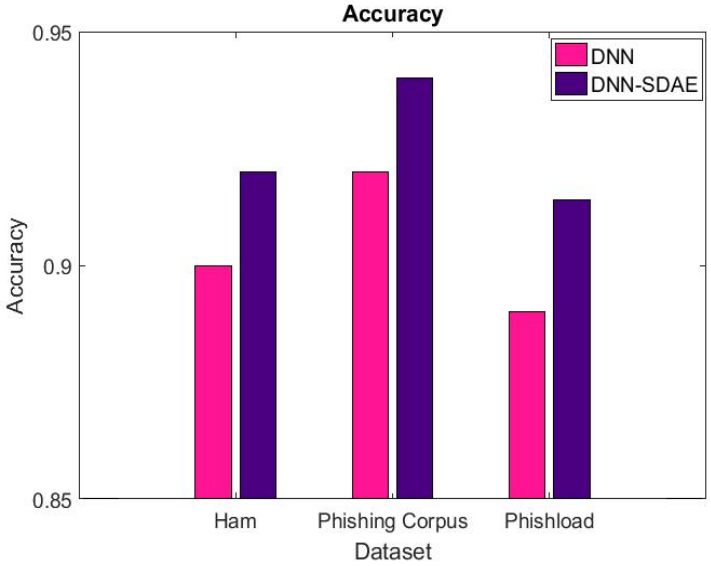

Fig.2: Comparison of Accuracy

Fig. 2 shows the comparison of accuracy between DNN and DNN-SDAE based phishing URL detection for Ham, Phishing Corpus and Phishload datasets. The datasets are taken in X-axis and the accuracy is taken in Y-axis. For Ham dataset, the accuracy of DNN-SDAE based phishing URL detection is 2.22% greater than DNN based phishing URL detection. Similarly, for Phishing Corpus dataset, the accuracy of DNN-SDAE based phishing URL detection is 2.17% greater than DNN based phishing URL detection. The accuracy of DNN-SDAE based phishing URL detection is 2.7% greater than DNN based phishing URL detection for Phishload dataset. From this analysis, it is proved that the proposed DNN-SDAE based phishing URL detection has high accuracy than DNN based phishing URL detection.

4.2. Precision

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

$$Precision = \frac{True\ Positive}{True\ Positive + Fals\ \ Positive} \quad (9)$$

Table 2 shows the comparison between DNN and DNN-SDAE in terms of precision for different datasets.

Table 2: Comparison of Precision

Datasets	DNN	DNN-SDAE
Ham	0.88	0.91
Phishing Corpus	0.89	0.916
Phishload	0.87	0.90

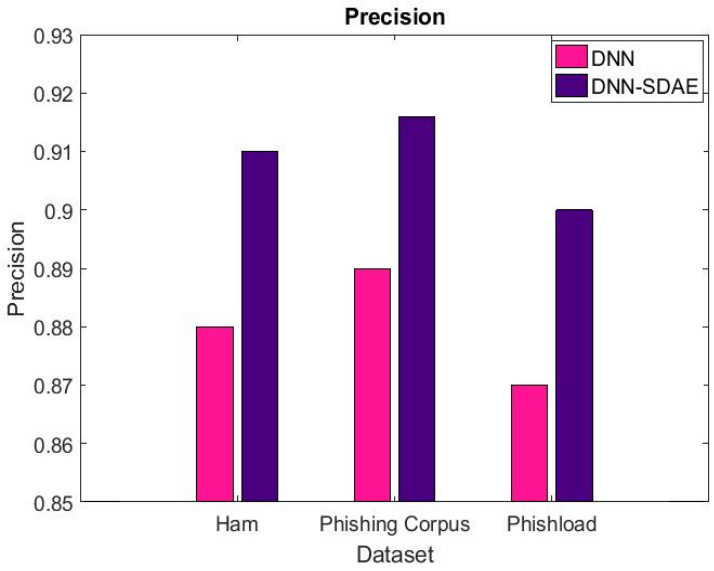

Fig.3: Comparison of Precision

Fig. 3 shows the comparison of precision between DNN and DNN-SDAE based phishing URL detection for Ham, Phishing Corpus and Phishload datasets. The datasets are taken in X-axis and the precision is taken in Y-axis. For Ham dataset, the precision of DNN-SDAE based phishing URL detection is 3.41% greater than DNN based phishing URL detection. Similarly, for Phishing Corpus dataset, the precision of DNN-SDAE based phishing URL detection is 2.92% greater than DNN based phishing URL detection. The precision of DNN-SDAE based phishing URL detection is 3.45% greater than DNN based phishing URL detection for Phishload dataset. From this analysis, it is proved that the proposed DNN-SDAE based phishing URL detection has high precision than DNN based phishing URL detection.

4.3. Recall

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

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (10)$$

Table 3 shows the comparison between DNN and DNN-SDAE in terms of recall for different datasets.

Table 3: Comparison of Recall

Datasets	**DNN**	**DNN-SDAE**
Ham	0.87	0.9
Phishing Corpus	0.88	0.915
Phishload	0.88	0.92

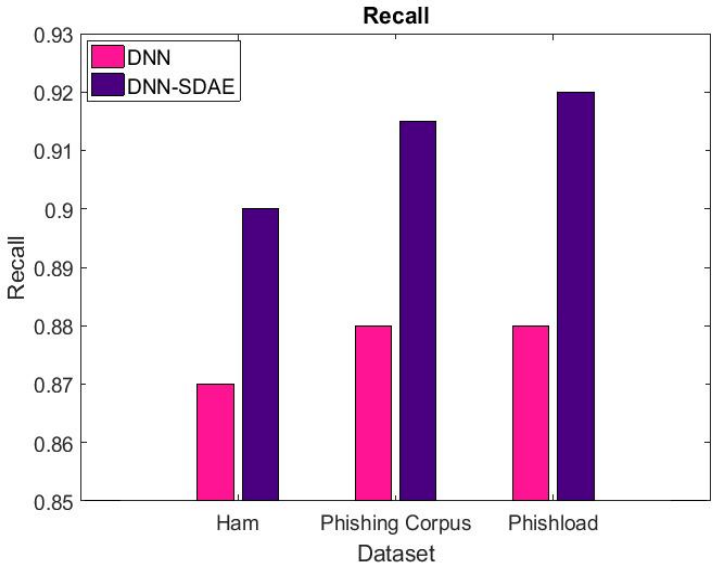

Fig.4: Comparison of Recall

Fig. 4 shows the comparison of recall between DNN and DNN-SDAE based phishing URL detection for Ham, Phishing Corpus and Phishload datasets. The datasets are taken in X-axis and the recall is taken in Y-axis. For Ham dataset, the recall of DNN-SDAE based phishing URL detection is 3.45% greater than DNN based phishing URL detection. Similarly, for Phishing Corpus dataset, the recall of DNN-SDAE based phishing URL detection is 3.98% greater than DNN based phishing URL detection. The recall of DNN-SDAE based phishing URL detection is 4.55% greater than DNN based phishing URL detection for Phishload dataset. From this analysis, it is proved that the proposed DNN-SDAE based phishing URL detection has high recall than DNN based phishing URL detection.

4.4. F-measure

F-measure is computed as the mean of precision and recall. It is calculated as,

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (11)$$

Table 4 shows the comparison between DNN and DNN-SDAE in terms of f-measure for different datasets.

Table 4: Comparison of F-measure

Datasets	DNN	DNN-SDAE
Ham	0.87	0.905
Phishing Corpus	0.885	0.915
Phishload	0.878	0.904

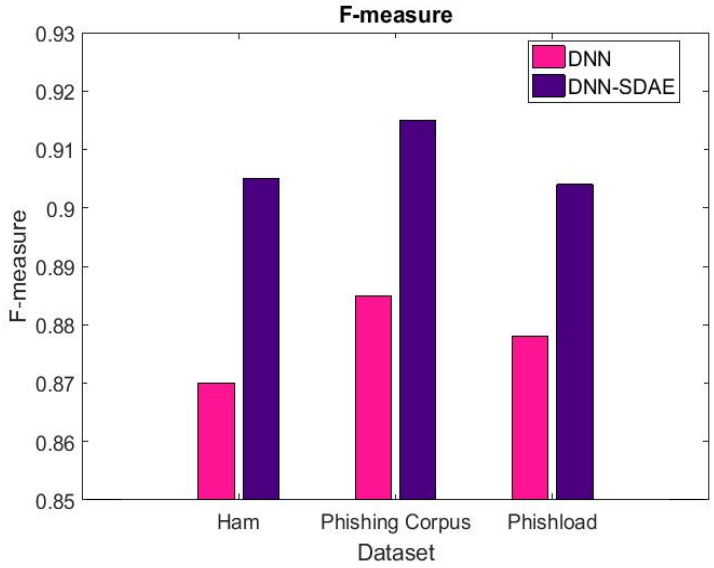

Fig.5: Comparison of F-measure

Fig. 5shows f-measure value of DNN and DNN-SDAE based phishing URL detection for Ham, Phishing Corpus and Phishload datasets. The datasets are taken in X-axis and the f-measure is taken inY-axis. For Ham dataset, the f-measure of DNN-SDAE based phishing URL detection is 4.02% greater than DNN based phishing URL detection. Similarly, for Phishing Corpus dataset, the f-measure of DNN-SDAE based phishing URL detection is 3.39% greater than DNN based phishing URL detection. The f-measure of DNN-SDAE based phishing URL detection is 2.96% greater than DNN based phishing URL detection for Phishload dataset. From this analysis, it is proved that the proposed DNN-SDAE based phishing URL detection has high f-measure than DNN based phishing URL detection.

5. Conclusion

In this paper, DNN with SDAE is introduced for efficient phishing URL detection. A feature extractor extracts URL-based features, web-based features and domain-based features. The extracted features are reconstructed by SDAE which returns high-level features. These features are acted as input to DNN which selects the most important features and classifies the URLs as legitimate or phishing URLs by using softmax classifier. The SDAE returns sparse features which increase the classification accuracy of DNN. The experimental results prove that the proposed DNN-SDAE has high accuracy, precision, recall and f-measure for Ham, Phishing Corpus and Phishload datasets.

References

[1]. Basnet, R.B.; Mukkamala, S.; Sung, A.H. “Detection of Phishing Attacks: A Machine Learning Approach”. *Soft Comput. Appl. Ind.***2008,** *226*, 373-383. doi: 10.1007/978-3-540-77465-5_19.

[2]. Chin, T.;Xiong, K.; Hu, C."Phishlimiter: A Phishing Detection and Mitigation Approach Using Software-Defined Networking". *IEEE Access*, **2018,***6*, 42516-42531.doi: 10.1109/ACCESS.2018.2837889.

[3]. Sumathi, K.; Sujatha, V."Deep learning based phishing attack detection". **2019**. *In Progress.*

[4]. Du, B.; Xiong, W.; Wu, J.; Zhang, L.; Zhang, L.; Tao, D.; "Stacked convolutional denoising auto-encoders for feature representation". *IEEE trans.cybern.***2016**, *47*, 1017-1027.doi: 10.1109/TCYB.2016.2536638.

[5]. Masci, J.; Meier, U.;Cireşan, D.;Schmidhuber, J.; "Stacked convolutional auto-encoders for hierarchical feature extraction". *Int. Conf.Artif. Neural Netw.*Springer, Berl.,Heidelb., **2011**, 52-59.doi: 10.1007/978-3-642-21735-7_7.

[6]. Hamid, I. R. A.;Abawajy, J."Hybrid feature selection for phishing email detection. *Intern. Conf. Algorithms Archit. Parallel Process".,Springer, Berl., Heidelb.***2011,**266-275. doi: 10.1007/978-3-642-24669-2_26.

[7]. Shekokar, N. M.; Shah, C.; Mahajan, M.;Rachh, S."An ideal approach for detection and prevention of phishing attacks". *ProcediaComput. Sci.* **2015,** *49,* 82-91. doi: 10.1016/j.procs.2015.04.230.

[8]. Smadi, S.;Aslam, N.; Zhang, L."Detection of online phishing email using dynamic evolving neural network based on reinforcement learning". *Decis. Support Syst.***2018,** *107,* 88-102. doi: 10.1016/j.dss.2018.01.001.

[9]. Hota, H.S.; Shrivas, A.K.; Hota, R."An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique". *ProcediaComput. Sci.***2018,** *132,* 900-907. doi: 10.1016/j.procs.2018.05.103.

[10]. Zhu, E.; Liu, D.; Ye, C.; Liu, F.; Li, X.; Sun, H."Effective Phishing Website Detection Based on Improved BP Neural Network and Dual Feature Evaluation". *IEEE Intl Conf Parallel &Distrib. Process. Appl., Ubiquitous Comput. Commun., Big Data & Cloud Computing, Social Computing & Networking, Sustainable Comput. Commun.***2018,**759-765. doi: 10.1109/BDCloud.2018.00114.

[11]. Zabihimayvan, M.; Doran, D."Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection". *arXiv preprint arXiv:1903.05675***2019**.doi: 10.1109/BDCloud.2018.00114.

[12]. Chiew, K.L.; Tan, C.L.; Wong, K.; Yong, K.S.; Tiong, W. K."A new hybrid ensemble feature selection framework for machine learning-based phishing detection system". *Inf. Sci.***2019,** *484*, 153-166. doi: 10.1016/j.ins.2019.01.064.

[13]. Abutair, H. Y.;Belghith, A. "Using Case-Based Reasoning for Phishing Detection". *ProcediaComput. Sci.*, **2017,***109*, 281-288.doi: 10.1016/j.ins.2019.01.064.

Author's Biography

KothandanSumathi, Ph.d Research scholar, CMS College of Computer Applications, CMS College of Science and Commerce, Chinnavedampatty, Coimbatore, Tamilnadu, India.

KothandanSumathi 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.

VijayanSujatha,Dean-Administration, CMS College of Computer Applications, CMS College of Science and Commerce, Chinnavedampatty, Coimbatore, Tamilnadu, India.

VijayanSujathahas 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.

How to Cite

KothandanSumathi and V. Sujatha, "Deep Neural Network with Stacked Denoise Auto Encoder for Phishing Detection", *International Journal of Machine Learning and Networked Collaborative Engineering,* Vol. 03, No. 2, 2019, pp. 114-124.

doi : https://doi.org/10.30991/IJMLNCE.2019v03i02.005.