Autoencoder based Architecture for Mitigating phishing URL attack in the Internet of Things (IoT) using Deep Neural Networks

S. B. Gopal
Department of ECE
Kongu Enginerring College
Erode, India
s.b.gopalece@gmail.com

C. Poongodi
Department of IT
Vivekanandha College of Engineering for Women
Namakkal, India
poongodi321@yahoo.co.in

D. Nanthiya
Department of CT-UG
Kongu Enginerring College
Erode, India
nanthiya.ctug@kongu.edu

T. Kirubakaran
Department of ECE
Kongu Enginerring College
Erode, India
kirubaraji26@gmail.com

D. Logeshwar
Department of ECE
Kongu Enginerring College
Erode, India
logeshdm123@gmail.com

B. Kulavishnu saravanan
Department of ECE
Kongu Enginerring College
Erode, India
ksvishnube@gmail.com

***Abstract*— One of the serious emerging threats on the internet is a Phishing URL attack, wherein the attackers try to steal the login credentials like username, passwords, or even bank details of the user by creating fake websites then, get into the actual website then try to corrupt or delete the information. According to a survey conducted in 2016 at least two-thirds of people were exposed to these phishing attacks so, these types of attacks must be taken seriously to avoid losses occurred by these cyber threats. There are some existing defense mechanisms like a firewall, but it is not effective in preventing different forms of attacks, since it works on a set of predefined rules, it identifies and blocks only those predefined attacks. Defending mechanism that itself able to learn the possibility of attacks so that, it not only identifies and blocks the existing forms of attacks but also any other forms of future attacks must be needed. Our proposed method uses Deep Neural Networks (DNN) model combined with autoencoders, to detect and block phishing websites in the network layer. The model is used to feature reduction and trained with the help of a dataset received from autoencoder, which consists of a set of legitimate websites and phishing websites. This architecture shows the accuracy of about 98.18% for Multiclassification Dataset. Since the memory, availability in sensor nodes is less only limited processing of data can be done security algorithms are implemented in network layers.**

Keywords— ANN, Autoencoders, Deep learning, Phishing, IoT.

I. INTRODUCTION

The world is evolving day by day and everything has become automatic. Automation has evolved staggeringly particularly once the introduction of a web of Things (IoT). IoT may be a tool that connects each object with one different, helps to exchange and manage info with different devices while not human interferences. Variety of web users is increasing day by day. It is expected that the number of web users is going to be inflated by 974 million by the top of 2025.

A phishing attack aims to carry out fraudulent activities using information stolen from the victims. It has a variety of methods for stealing information shown in Fig. 1. The steps involved are as follows

- Deciding the target
- Compose email, messages, or pretend URLs
- Attack and install malware on a system
- Gather the desired information like word, username, bank credentials
- Try and access the initial web site and corrupt the info

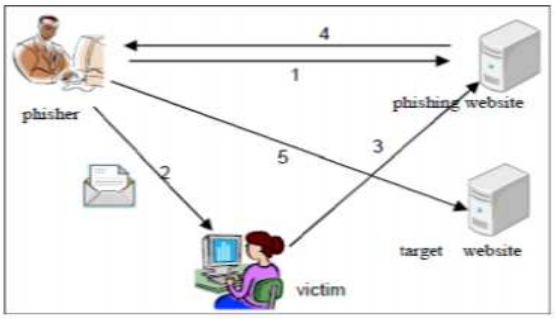


Fig. 1. Phishing attack detection techniques

There are various types of phishing attacks, but their main motives are the same. Depending upon the number of targets and the mechanism used to collect the data there are different kinds of attacks [5][12]. They are

- Deceptive Phishing
- Spear Phishing
- Whaling
- Pharming

II. LITERATURE SURVEY

A. Research Gap

Evolving digital transformation has made a problem in security of networks worldwide. With help of evolving technologies phishers tend to get access to the organization private information without their knowledge. In our project we have proposed an Autoencoder based prototype to mitigate the phishing attacks through Deep Neural Networks.

978-1-6654-8094-9/22/$31.00 ©2022 IEEE

B. Survey Table

TITLE OF THE PAPER	NAME OF THE JOURNAL, VOL NO, YEAR	METHOD
Detection of Phishing in Internet of Things Using Machine Learning Approach	2021 International Journal of Digital Crime and Forensics Volume:13, Issue:2, March - April 2021	Random Forest, SVM, Logistic Regression
Detecting Internet of Things attacks using distributed deep learning	2020 Journal of Network and Computer Applications 163 (2020) 102662	CNN, LSTM.
Phishing Attacks Detection using Machine Learning Approach	IEEE Transactions on Dependable and Secure Computing (Volume:15, Issue: 6, Nov.-Dec. 1, 2018)	PCA, Random Forest.
A Phishing Detection System based on Machine Learning	2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)	SVM, Decision Tree, Logistic Regression
Machine Learning Inspired Phishing Detection (PD) for Efficient Classification and Secure Storage Distribution (SSD) for Cloud-IoT Application	2020 IEEE Symposium Series on Computational Intelligence (SSCI)	SVM, KNN Random Forest.

III. EXISTING METHOD

In the existing method, PCA (Principle Component Analysis) and RF (Random Forest) is used to classify whether the data is phishing or legitimate. Here PCA is used for feature selection and then RF works as a classifier. Here the dataset is selected which contains both phishing and legitimate data. After the selection of the dataset, it is preprocessed, and feature selection is done using PCA. Then the selected dataset is given as input to the Random Forest to classifiy whether the given data is phishing or legitimate. As shown in Fig. 2. depicts the working of an existing methodology with PCA and RF.

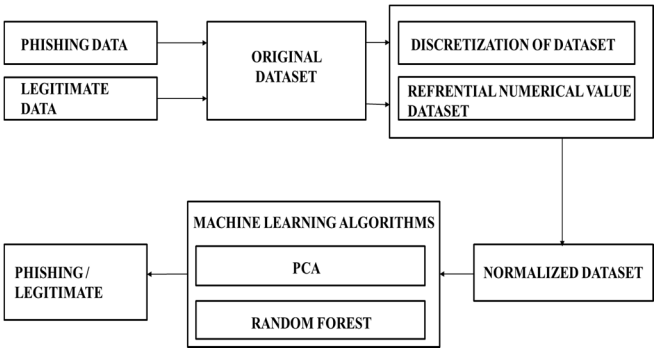


Fig. 2 Architecture of Existing Methodology

A. Principle Component Analysis

PCA, Principal Component Analysis is a technique that is used to decrease the dimension of the given data. The main aim of PCA is to make bigger variables lower size. Firstly, PCA finds a design or interconnection among various characteristics of datasets. After figuring out the important interconnection among the different variables a conclusion is made to resize the data so that some of the important information are not being lost. This process is important while finding solution to problems that use data sets whose dimensionality is high. The process involved in PCA,

- Uniformity
- Determination of Co-Variance Matrix
- Determine The Eigen Value and Eigen Vector
- Evaluating The Principal Components
- Dimentionality Reduction

B. Random Forest

RF is a supervised learning technique. It consists of large number of decision tree where the input data set is splitted into subsection and given to every tree. From the outcome of each decision tree the performance of the model is predicted, and the RF categorize the end output. The work plan of Random Forest is,

- Among the training set, pick a irregular K data value.
- Construct the decision trees which are related with the picked data values that are considered as subsections.
- Number of decision trees to be constructed is chosen.
- Now figure out the outcome of every decision tree and for the subsection that wins the maximum count of votes will be given a new data value.

IV. PROPOSED METHOD

Phishing is an identity theft that makes use of an internet platform to gain individuals' confidential data to illegally access the accounts. The number of victims of phishing attacks was increasing dramatically. Since the attackers continue developing new phishing techniques [9],[8] it is hard to find and block these attacks. We had proposed a method that uses the ANN model to identify and blocks phishing attacks.

To train the model we need a dataset. We had downloaded a dataset URL dataset(ISCX-URL2016) from UNO. It is a multi-classification dataset that contains different types of phishing URLs. It contains over 35,000 benign types of URLs collected from Alexa, around 12000 Spam URLs which is

collected from WEBSPAM-UK2007, 10,000 phishing URLs collected from Open phish, around 11,500 malware URLs collected from DNS-BH, and more than 45,450 URLs belong to the Defacement URL category.

A. Artificial Neural Network

Artificial Neural Networks (ANN) are multi-layered neural design that include of the entry or input, hiding, and end or output layers. By increasing the number of hidden layers, the model can be made substantially more complex[7]. It performs some manipulations that are non-linear at every layer to reflect the input data to a distinct feature space, and draw a critical decision border to divide the classes[1]. It only transforms its inputs non-lyrically between vectors so that it is projected into a higher dimension space, which allows it to be linearly divided. Any complex function can be modeled by ANN.

B. Autoencoder

Autoencoder is a type of ANN used to learn a set of datasets, usually for ignoring the noise by reducing the dimensions of inputs in addition to the reduction side, the autoencoder will also learn a re-building side in which the exact input is generated from the reduced encoded part[4]. Their main aim is to reconstruct the input [2] based on encoded learning Autoencoder has three main parts are encoder, code, and decoder shown in Fig. 3.

The encoder part captures the important features in training data, sufficient for approximate reconstruction. The maximum reduction is done in the bottleneck layer. The decoder part is the same as the reverse of the encoder part[12].There are seven types of encoders, denoising encoder, sparse encoders, deep encoders, contractive encoders, under complete encoders, convolutional encoders, and variational encoders.

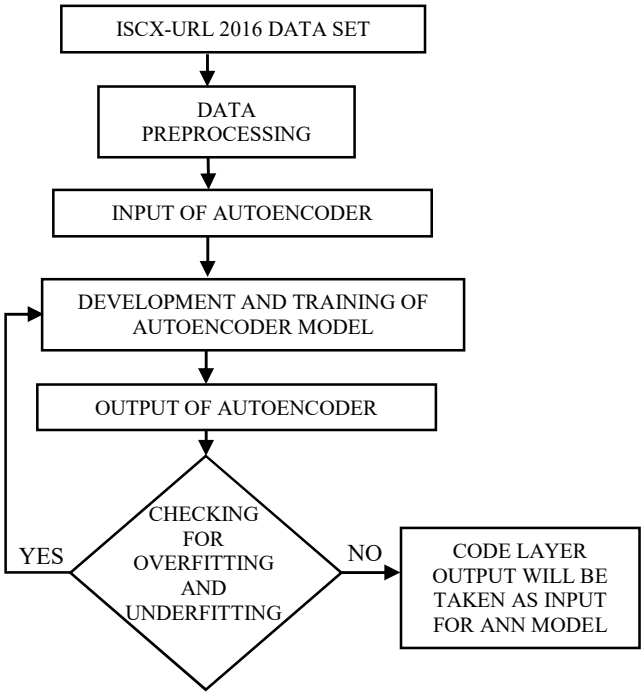


Fig. 3. Flowchart for Autoencoder Model

C. Loss Function

Loss or error function is a method of computing how good our algorithm fits with our dataset. Loss function is one of the significant features of deep neural network. As the name says it is used to predict the error in the neural network. Loss function also helps to compute the difference between the obtained value and the actual value.

Types of Loss Function:

- Binary Cross Entropy
- Hinge Loss
- Square Hinge Loss
- Kullback_Leiber_Divergence
- Categorical Cross Entropy

D. Activation Function

Activation functions (AFs) are used in ANN to compute some difficult calculations in the middle or hidden layer and then pass on the obtained result to the end or final layers. The work of these functions is to add some properties to the neural network that are non-linear, to change signals of a node that are linear to signals that are non-linear in nature [5].

E. Optimizers

It is very significant to vary the weights of the training model to ensure the predicted output as correct as possible. Hence, we use Optimizers which keeps on updating the training model in response to the result of loss function thereby helping the model to give accurate results efficiently. Some examples of optimizers are Adagrad, Adam, Sgd, Adadelta, Rmsprop and Nadam.

F. Work Flow

Initially, we had downloaded the dataset URL dataset(ISCX-URL2016) from UNO and we had done pre-processing in it. Pre-processing is usually done to clean by removing or replacing the missing values in the dataset [15] and to convert the raw data into a machine-understandable form shown in Fig. 4.

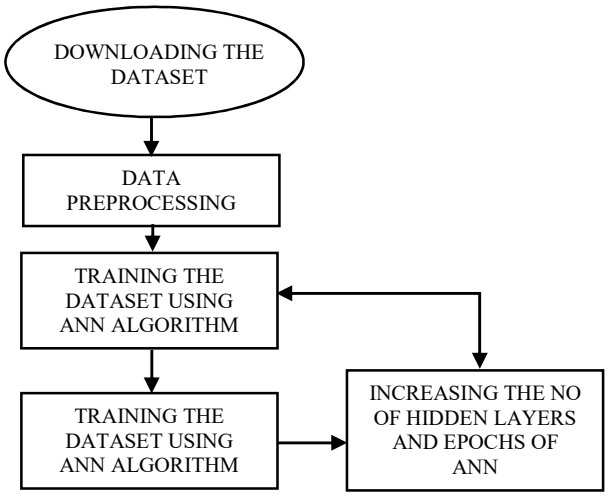


Fig. 4. Workflow of Proposed work

V. Result Analysis

After building the model it is important to evaluate the quality of the model. There are various features to estimate

and test the productivity of the ANN model. They are accuracy, precision, recall, and F1-score.

1. Accuracy

Accuracy is used as a metric to justify the model performance [10], which can be defined, as:

$$\textbf{Accuracy} = \frac{No.of\ correctly\ predicted\ variables}{Total\ no.of\ tested\ variables} * 100 \quad \ldots. (1)$$

2. Precision

Precision is calculated as the ratio of correctly predicted positive observations to the total number of predicted positive observations.

$$\textbf{Precision} = \frac{True\ Positive}{Total\ no.of\ positive\ prediction} \quad \ldots. (2)$$

3. Recall

The recall is calculated as the ratio of correctly predicted positive observations to all observations in the actual class.

$$\textbf{Recall} = \frac{True\ Positive}{No.of\ Actual\ positives} \quad \ldots. (3)$$

4. F1 – Score

F1 score balances between accuracy and precision. It is the harmonic mean of accuracy and precision. A high F1 score means that our model has low false positives and low false negatives, and so our model had to identify real threats and it was not disturbed by false alarms.

$$\textbf{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad \ldots. (4)$$

A. Performance Metrics With Autoencoders By Varying Epochs

Further, to improve the performance metrics we had implemented an autoencoder in our model. Then we had calculated the performance metrics. We had trained our model for 50 epochs, 100 epochs, 200 epochs, 400 epochs, 500 epochs and evaluated the performance metrics. We had also shown the bar graph representation of the obtained results as shown in Fig. 5. Our model performs effectively at training it at 500 epochs where we had obtained good results in performance metrics where we had obtained an accuracy of about 98.8%. By implementing autoencoders, we can see that our accuracy was increased by around 3%.

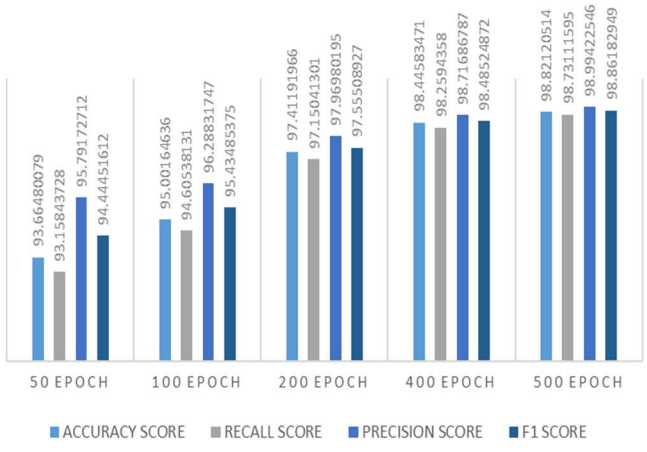


Fig. 5. Bar graph representation of varying epochs using autoencoders

B. Performance Metrics By Varying Hidden Layers With Auto Encoders

To improve the performance metrics further, we had also evaluated the model by implementing autoencoders and then varying the number of hidden layers. Firstly, we had trained our model for one hidden layer, we had increased the layer one by one and evaluated the performance metrics. We had also shown the bar graph representation of the obtained results as shown in Fig. 6. Each time when we increase the number of hidden layers, we can see that our model shows a gradual increase in performance metrics since the number of learnings increases when we increase the number of hidden layers. Our model performs effectively when we use five hidden layers where we had obtained an accuracy of about 96 percent our accuracy was improved by 4 percent.

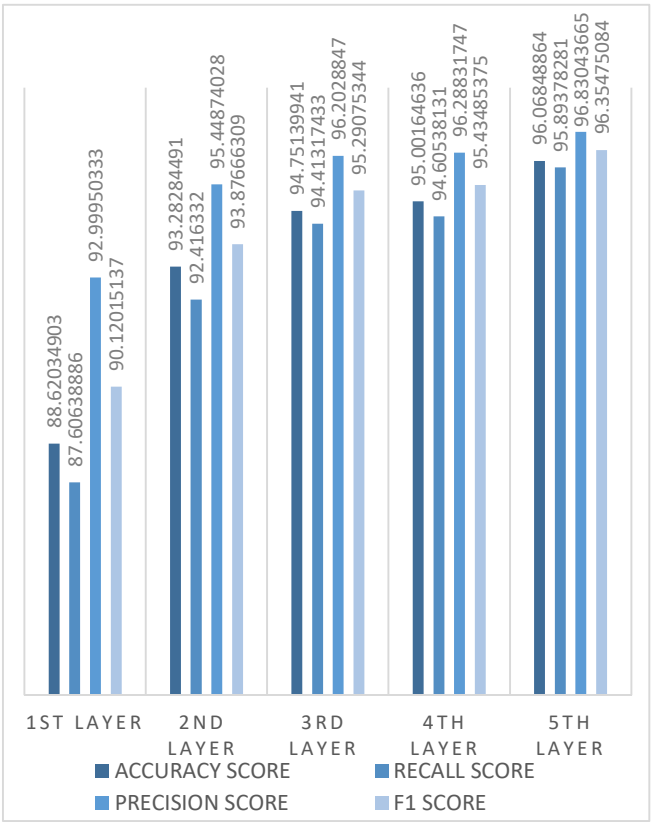


Fig. 6. Bar graph representation of varying hidden layers using autoencoders

C. Comparison By Varying Optmizer With Loss Functions

By fixing number of hidden layers as Four and Epochs as 500, the comparison has been done over the performance with various optimizers as shown in Table 1.

TABLE I
PERFORMANCE METRICS BY VARYING OPTIMIZER WITH BINARY CROSS ENTROPY LOSS FUNCTION

Optimizer	Accuracy	Recall	Precision	F1-Score
Adam	98.89	98.79	98.95	98.87
Adagrad	58.14	55.28	75.52	62.10
Adamax	98.22	98.00	98.44	98.22
Adadelta	16.55	12.30	56.25	15.12

SGD	82.65	80.14	86.30	82.59
Nadam	98.81	98.66	98.85	98.75
RMSprop	98.27	98.12	98.35	98.24

D. Comparison of Proposed Work with Existing Work

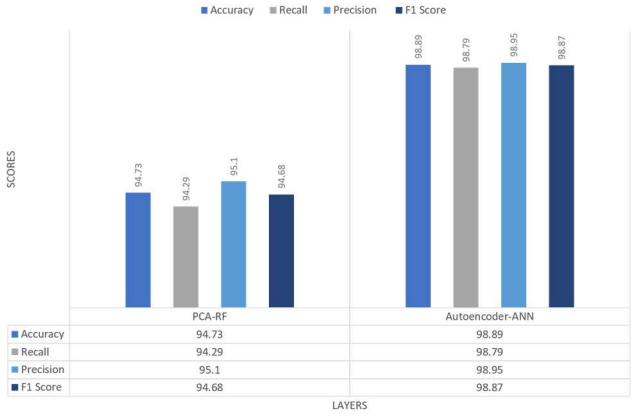


Fig. 7. Bar graph representation of varying epochs using autoencoders

The comparison difference between the parameters of the existing and proposed work shown in fig. 7. The parameters such as accuracy, recall, precision and F1 score are analysed between the principle component algorithm with random forest and Autoencoder with ANN algorithm. Based on this analysis, that all parameters have been improved in the Autoencoder-ANN algorithm in the Proposed method. So that we have obtained a maximum score in all parameters than the existing work. Hence, we inferred that all parameters in our proposed model was improved by 3.2% in all the cases.

VI. Conclusion and Future Work

In this paper, we had proposed a method to secure IoT by blocking the phishing URLs directed to client's IoT devices by using ANN models and autoencoders. A publicly available dataset has been used to train the ANN model [6]. We had split 80 percent of the dataset to train the model and we had used the remaining datasets to evaluate the performance of the model. We observed that ANN exceeds in performance with autoencoders in comparison with ANN without autoencoders. By improving the number of hidden layers and epochs, we had able to improve the accuracy of our model.

We had obtained an accuracy of about 95.6% and by implementing auto encoders into our model our accuracy was improved to 98% which indicates that our accuracy improved by around 3%.

Our model performs effectively when we use four hidden layers where we had obtained an accuracy of about 98.79 % in Layer 4 with Epoch 500 when compared with other 3 layers. Also, we observed that with the use of adam optimizer with binary cross entropy loss function we improved our accuracy of the model to about 98.89%.

In the future, we can train our model with more real-time datasets and evaluate their performance in various environments.

References

[1] Nirmal, K., B. Janet, and R. Kumar, Analyzing and eliminating phishing threats in IoT, network and other Web applications using iterative intersection. Peer-to-Peer Networking and Applications, 2020: p. 1-13.

[2] Gupta, B.B., et al., Fighting against phishing attacks: state of the art and future challenges. Neural Computing and Applications, 2017. 28(12): p. 3629-3654.

[3] Naaz, S., Detection of Phishing in Internet of Things Using Machine Learning Approach. International Journal of Digital Crime and Forensics (IJDCF), 2021. 13(2): p. 1-15.

[4] Rao, R.S., T. Vaishnavi, and A.R. Pais, PhishDump: A multi-model ensemble based technique for the detection of phishing sites in mobile devices. Pervasive and Mobile Computing, 2019. 60: p. 101084.

[5] Parra, G.D.L.T., et al., Detecting Internet of Things attacks using distributed deep learning. Journal of Network and Computer Applications, 2020. 163: p. 102662.

[6] Sahu, K. and S. Shrivastava, Kernel K-means clustering for phishing website and malware categorization. International Journal of Computer Applications, 2015. 111(9).

[7] Sameen, M., K. Han, and S.O. Hwang, Phishhaven—an efficient real-time ai phishing urls detection system. IEEE Access, 2020. 8: p. 83425-83443.

[8] Feng, J., et al., Web2Vec: Phishing Webpage Detection Method Based on Multidimensional Features Driven by Deep Learning. IEEE Access, 2020. 8: p. 221214-221224.

[9] Yang, P., G. Zhao, and P. Zeng, Phishing website detection based on multidimensional features driven by deep learning. IEEE Access, 2019. 7: p. 15196-15209.

[10] Ravi, R., A performance analysis of Software Defined Network based prevention on phishing attack in cyberspace using a deep machine learning with CANTINA approach (DMLCA). Computer Communications, 2020. 153: p. 375-381.

[11] Prasanna Kumar, K.R. & Kousalya, K. "Amelioration of task scheduling in cloud computing using crow search algorithm", Neural Comput & Applic (Online: Feb 2019 Issue: May 2020).

[12] S. B. Gopal, C. Poongodi, M. Joseph Auxilius Jude, S.Umasri, D. Sumithra, P.Tharani, "Minimum Energy Consumption Objective Function For RPL In Internet Of Things", International Journal Of Scientific & Technology Research Volume 9, Issue 01, January 2020.

[13] Nanthiya, D, Keerthika, P, Gopal, S.B., "Anticipation of wormhole attacks by selective routing in wireless sensor networks", International Journal Of Scientific & Technology Research Volume 9, Issue 01, March 2020.

[14] S.B.Gopal, C Poongodi , "Mitigating DoS attacks in IoT using Supervised and Unsupervised Algorithms–A Survey ",IOP Conference Series: Materials Science and Engineering, 2021,Kongu Engineering College.

[15] Gopal, S.B., Poongodi, C., Nanthiya, D," Enhancement of routing protocol for low power lossy network for internet of things", Proceedings of IEEE International Conference on Intelligent Computing and Communication for Smart World, I2C2SW 2018, 2018, pp. 354–356.