EBONN: AN ENHANCED BAYESIAN OPTIMIZED NEURAL NETWORK FOR CLASSIFICATION OF PHISHING ATTACKS

N.SWAPNA GOUD[1],DR. ANJALI MATHUR[2]

[1]Research scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.

[2]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

Corresponding Author: swapnagoudcvsr@gmail.com, anjdemo@gmai.com

ABSTRACT

Phishing attacks became most common cyber security attack in the digital platforms. To identify these types of attacks using URL analysis, the proposed research has identified the important features and then to design a customized neural network, it has enhanced Bayesian optimizer by adding layers and estimator values to find the best parameters that suits for the given dataset. Previous works has focused on implementation using either traditional or pre-trained approaches which has achieved least accuracy with approximate values of 65.25%, so to find the solution for this problem, the proposed research defined an algorithm which defines the objective function to maximize the target values of the estimators. It also implements stratified validation to find the values at iteration and picks the one with best value as output. The model has parameterize all the estimators associated with the neural network and found that it works with an accuracy of 99.5% for training data and with 99.7% for validation data.

Keywords: *Exponential Linear Function, Activation Function, Optimization, Objective Function, Batch Normalization, Pre-Trained Model.*

1. INTRODUCTION

Most of the previous researchers implemented the detection of phishing websites using either traditional machine learning approaches or deep learning approaches and has achieved good accuracy. The implementations of these algorithms have used brute force techniques to design the neural network. Even though they got good accuracies but these systems doesn't have any standard approach to decide values that fit each and every estimator. So, the proposed research wants to find the best parameters for the customized neural network. The following section discusses about the components of the neural networks in detail.

1.1. Neural Network:

It is biological based technique which updates itself in every iteration by learning the features from previous iterations and observations made on the weights. Many real time applications associated with image processing, speech recognition, and natural language processing are using pre-trained models for implementation of deep learning techniques. The type of neural network depends on the number of hidden layers implemented by the model. If the model implements single hidden layer then it is known as "Shallow NN" as shown in figure 1. If the model implements more number of hidden layers then it is known as "Deep NN".

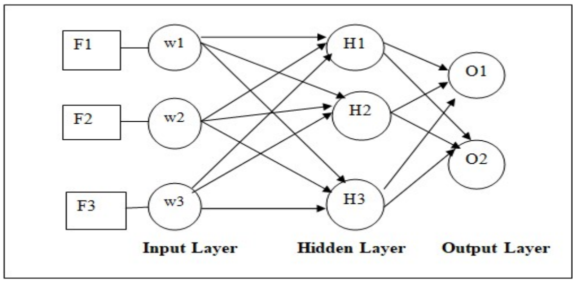


Figure 1: Shallow Neural Network- Fully Connected in Nature

The basic component of the neural network is “neuron”, it receives the information and it has the capability to perform all types of mathematical computations to generate necessary output based on the requirement of the model. There are three types of neurons, a. input neuron, and it gathers necessary information in the numerical form from the real world. b. hidden neuron, it is considered as the vital neuron, because it has to distinguish the simple features and more informative features, then it has to map the features for finding the correlations between them. c. output neuron, it has acts the decision maker to predict the final class label and prints the output either in numerical or binary format based on the type of classification.

1.2. Types of Layers in Neural Networks

The efficiency in terms of accuracy and loss validation of the neural network depends on the type of layers, their combinations and number of layers implemented in the model. Basically, there are 6 types of layers present in the NN as shown in figure 2.

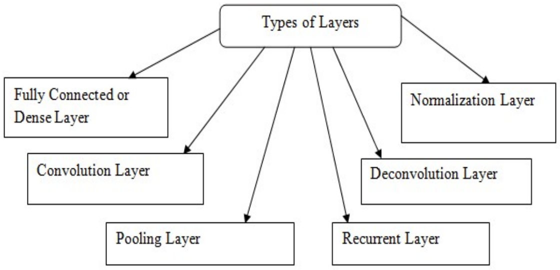


Figure 2: Categorization of Layers in Networks

Fully Connected Layers connects every neuron with every other possible neuron in next layer. The implementation and computational cost of these networks are very high at the same time the scalability of this is very low. The major advantage of this lies in optimizing its estimator, which are very less i.e., three in number. These types of layers are generally applied in image classification with huge amounts of data. The best layer to perform the feature extraction is “Convolution Layer”. It has the capability to scan and operate on N-pixels simultaneously by defining the kernel size. To determine whether a filter applied pixels forms a feature or not, it applies probability and performs patches on them. The real time application for this is “Image Recognition”. The pooling layer major goal is to reduce the size of the image representation by reducing the computation and number of trainable parameters. While down sampling the layers, it summarizes the features in each patch and decides whether to consider or ignore these features. The transposed version of convolution layer is deconvolution, which performs the up sampling technique on the features. These layers rarely implemented when the model wants to increase the quality of images or to remove the noise in the images caused by various external factors. A layer which has the looping capability, i.e., to take feedback from output by analyzing the computed inputs and recompute the values to optimize the model. When the values of the pixels vary with huge difference because of the intensity levels then it is better to standardize all the values by applying “Normalization Layer”.

1.3. Activation Functions:

The task of the activation function is to standardize all the input values. This is the major deciding factor in neural network, which explores the importance of an input feature given to a neuron. Based on the importance, it either activates or deactivates the neuron. There are three types of activation function. Table 1 represents sub classification along with their representation.

Table 1: Categorization of Activation Function

S.No	Type	Sub Classification	Activation Function
1	Binary Step	-	f(input)= 0, if input <0 = 1,if input>=1
2	Linear	-	f(input)=input
3	Non-linear	Sigmoid	f(input)=$\frac{1}{1+e^{-(input)}}$
		Tanh	f(input)=$\frac{e^{input}-e^{-(input)}}{e^{input}+e^{-(input)}}$
		ReLu	f(input)= max(0,input)
		Leaky ReLu	f(input)=max(0.1*input, input)
		Parametric ReLu	f(input)=max(a*input, input), where a is slope value

		Softmax	f(input)=$\frac{e^{z_{input}}}{\sum_{i=1}^{n} e^{z_{output}}}$, n represents number of classes in multi classification problem, z_{input} represents input vector and z_{output} represents output vector
		Swish	f(input)=input*$\frac{1}{1+e^{-(input)}}$

1.4. Loss Functions:

Every neural network has an objective function associated with it. So, to evaluate the model whether it is good or bad, the network implements loss function in terms of mathematical generalization. Loss functions are categorized into classification losses and regression losses and these are illustrated in figure 3.

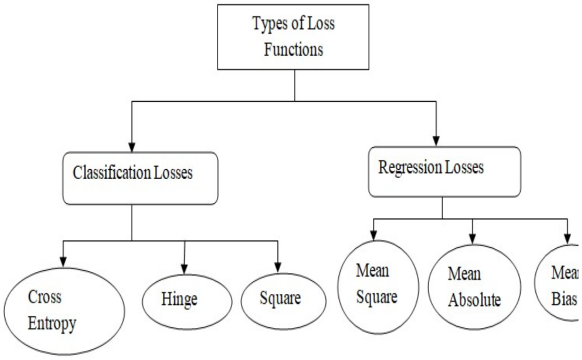


Figure 3: Classification of Loss Functions

2. LITERATURE SURVEY

The developer of this paper has explained the rising attacks in website phishing were several users, organizations and government bodies were unknowingly being the victims. The author had explained an attack where, the intruder gathers the confidential information by passing them some spoofed messages via emails or duplicate websites. These sites are usually developed in such a way that they looklike the original websites with similar content and themes which confuse the user and makes them think these as the original ones. This paper has presented an overview of various previously implemented research and have provided their corresponding data and outcomes in a tabular format for easier understanding. The literature in this paper have concentrated on finding out the best outcomes within the DL techniques implementation which are related to ML, DL, Hybrid training and scenarios depended on methodologies that are developed in an objective of identifying the phishing attacks [1].

According to the author, phishing attacks are some malicious sets of programs that illegally captures the users' data. These attacks, by the author, usually use email as their platform or develop a duplicate a webpage that resembles like the original site. This duplication would be so perfect in such a way that the users get confused and believe that the fake page is the original one. So,the objective lies around finding out such malicious websites, providing security to the users and other organizations in a process to overcome by reducing the victims to these phishing attacks. To maintain the users' confidentialitymany developers are now focusing on designing AI methods whereas, the developers of this paper worked on two different datasets having a massive collection of labelled data on websites with class specifications of phished and legitimate. Over the previous literature, ML, DL, and other selection of properties for identifying these attacks were determined in this paper. DT, kNN and RF model in ML techniques have been presented with an additional ReLU layer with DL were designed in this paper. Each specification regarding the evaluation metrics have been noted and a highest of 97.65% prediction rate was achieved [2].

The author states that the traditional ML methods only deploy classification for once that only focuses on maintaining the accuracy and depends mostly on that specific identification algorithm. To improve predictive performance, a system for merging the complimentary knowledge of several classifiers is necessary. The effectiveness of different NN methods for picking base classifiers is evaluated in this paper, which also models an aggregation technique for identifying phishing domains. RBF, GRBF, PNN, and HPNN were selected as basis categories depending on the preliminary findings. This method focuses on increasing the solo and joint quality of base learners for recognizing phishing websites. DeepEEviNNet is the recommended ensemble technique, is created by merging the results of base classifiers depending on their scores to reach a final judgment.The separation between the merger outcome obtained via DST and the fact determines the ideal value for each classifier. Furthermore, to identify the RBF cores and Gaussian kernels of the basic learners, a unique categorical grouping approach called WEFKM is developed [3].

Due to the high learning skills on large datasets and standard outcomes in many categorization tasks, ML and DL algorithms are becoming the most essential methods used to discover and inhibit phishing attempts. Earlier, two different types of functionality filtration strategies i.e., feature embedding based, and NLP extraction of properties were used in solitude. Unfortunately, because the scientists did not combine these characteristics, the results were not particularly impressive. In contrast to earlier research, this research proposed a method that incorporates extraction of features methodologies. We talked about how to integrate various characteristic retrieval strategies to get the most out of the data.This research offers hybrid DL methods that rely on LSTM and DNN strategies for identifying phishing consistent service locators and assesses the systems' effectiveness on phishing datasets. The introduced composite DL models combine text embedding with NLP characteristics, allowing them to leverage complex character relationships while also showing high-level NLP associations. The developed Light BGM model had gained 98.19 % of accuracy [4].

Phishing is a cybercrime that involves the theft of personal information from users. Customers, corporations, cloud services, and government agencies are all targets for the fake websites. Anti-phishing technologies predicated on equipment are commonly utilized, while software-based options are preferred due to affordability and administrative considerations. Contemporary phishing identification systems have no resolution for vulnerabilities like zero-day phishing assaults. To address these issues, a three-phase exploit surveillance system called the Phishing Assault Detection provided by Web Crawler was suggested, which uses a RNN framework to accurately identify malicious incidents. The proposed method additionally considers the input properties from the Web traffic, content on Web and URLs those rely on the identification of both phishing and original webpages. The developed model worked on both NB and RNN frameworks by gaining 90.29% and 92.26% of accuracies respectively [5].

For distinguishing malicious online pages from authentic internet pages, most present spoofing detection techniques use Bayesian categorization. These techniques are ideal when a dataset has a small amount of online pages and yield efficiency of 90%. The volume of the internet has grown dramatically in subsequent decades, and previous technologies have failed to deliver adequate precision for massive datasets. As a result, this study proposes a novel method for detecting phishing websites by looking for hyperlinks in the original content of the HTML webpage of the relevant domain. To recognize malicious online domains, the suggested methodology employs a feature representation with 30 characteristics.These characteristics are utilized to educate the supervised DNN machine with Adam optimizer to distinguish between false and legitimate websites. To distinguish between phishing and authentic websites, the suggested DL model with Adam Optimizer employs a Listwise technique. When contrasted to alternative classic ML algorithms such as SVM, Adaboost, and AdaRank, the suggested methodology performs admirably [6].

Cybercrime in web based commercial applications is fast expanding because of technological advancements. Invaders utilize a variety of tools and strategies to combat such assaults, including Phishing, Spyware, SQL Scripting, Ransomware, XSS, DoS, Session Hijacking, and Credential Reuse. Deception has become increasingly popular in the digital age which is a deceptive method of obtaining confidential material from unsuspecting consumers. Depending on the perpetrator's intended objective, these cyber-attacks can be used to hurt participants or whole businesses. The researching industry has come up with several phishing identification approaches to prevent these attempts.The developers had divided standard phishing identification techniques into 5 groups in this article: (1) DM, DL, and ML-Based Strategy, (2) Keyword Optimization Strategy, (3) URL Scan-Based Method, (4) Blacklisting-Whitelisting Process, and (5) Optical Similarity-Based Strategy. In addition, a comparative assessment of each class's strengths and flaws is presented [7].

The associative training methodology employing identified harmful URLs has been validated in the domain of DL, emphasizing the severity of phishing assaults, which is stressed by many organizations. Furthermore, because of the peculiarities of zero-day attacks, the DL-based solution primarily concentrated on adapting a categorization job via previous URL surveillance has a recollection restriction. An effective strategy that employs technical expertise is intriguing for modeling the characteristics of a zero-day spoofing assault in which URL identifiers are established and destroyed promptly. To infuse real-world constraints, we present a way for integrating DL and logical coded domain expertise.The developers have created neural and logical analyzers and suggested a mutual learning technique based on classic neuro-symbolic fusion for each element. Considering the difficult class-imbalanced situation, rigorous testing on 3 datasets containing varied URLs were conducted, and the approach demonstrated the maximum recall amongst recent DL algorithms. They have shown an optimizing the proportion between the neural and cognitive components improves recall by more than 3% when relative to previous approaches [8].

Deception on webpages is a type of cyber-attack that attacks internet users to obtain confidential material such as login passwords and financial details. Intruders deceive visitors by portraying a disguised webpage as real or reputable to collect vital information. Heuristic algorithms, blacklist or whitelist, and machine learning-based strategies have all been offered as remedies to phishing site attempts. The state-of-the-art strategies for detecting phishing websites using ML techniques are presented in this study, according to the author. Based on ML techniques, this study identifies remedies to the website's phishing challenge. The maximum of the methodologies investigated are based on classic ML techniques.The prominent ML approaches addressed in the literature are RF, SVM, NB, and Ada Boosting. In comparison to traditional ML algorithms, this assessment article highlights DL-based methods that function better at spotting phishing sites. Overfitting, low reliability, and the inefficacy of ML approaches in the absence of sufficient training data are among the problems discussed in this research [9].

According to the author of this research paper, the existing approaches to optical assessments face technological difficulties that prevent them from being productive and useful i.e., with low runtime overhead enough to be put to meaningful use. The programmers here created Phishpedia, a composite DL system, to handle two major technological issues in spoofing identification, namely (i) accurate identification of authenticity identities in webpage images, and (ii) correlating logo variations of the same identity. Both great precision and low execution latency are achieved with Phishpedia and, unlike many other techniques, the developed methodology here, does not necessitate any phishing examples to be trained on. Researchers

have conducted comprehensive tests using genuine phishing data, and the findings show that the developed system beats standard recognition techniques such as EMD, PhishZoo, and LogoSENSE in recognizing phishing sites reliably and rapidly [10].

In today's world, the Online world has evolved into a powerful tool for social connection. People's reliance on the internet platforms creates opportunities for deception. Phishing is a sort of criminality that involves stealing user's login information from digital sites like online payments, internet business, e-commerce, online classrooms, and virtual marketplaces, among others. Phishers create bogus websites that look just like the real thing and deliver phishing emails to lure consumers in. When an internet user sees a counterfeit homepage via spam, phishers steal their credentials. To identify malicious webpages, researchers have developed powerful tools like as blacklists, whitelists, and antivirus programs.Attackers are continually coming up with new ways to get around cyber defenses by exploiting people and network weaknesses. Adopting a data-driven method, this research proposes a data-driven methodology for detecting phishing webpages. To be more specific, the phishing sites are predicted using a multilayer perceptron, often known as a feed forward NN. The suggested framework has a learning accuracy of 95 % and a test accuracy of 93 % [11].

Table 2 discusses the previous mechanisms with their merits and demerits to identify the gaps involved in traditional approaches and to evaluate the performance.

Table 2: Existing Approaches using Deep Learning Techniques

S. No	Author	Algorithms	Merits	Demerits/ Future Work
1.	Abdul Basit	ML, DL, HL, scenario based.	This paper presented a collective study over the previous research literature.	Practical implementation was not done.
2.	Mesut TOĞAÇAR	Dt, k NN, RF and ReLU	Most of the performance calculations were made in finding out the classification and prediction of phishing sites.	Meta- heuristic optimization techniques could be implemented further.
3.	Priya S.	DeepEEviNNet	The proposed method worked on probabilistic constraints for gaining the phishing and actual weights of the learning cases to develop a combinatorial classifier.	Complex to compute. Should focus on deducing the dimensionality problem. These issues need to be dealt in future.
4.	Ozcan	DNN- LSTM and DNN – BiLSTM	The proposed methods can combine the next level features of NLP along with finding out the deeper connections among the characters.	In future, properties gained by word embeddings could be included to the current hybrid model.
5.	Rani Shinde	NB, RNN	The proposed method is said to gain highest precision and prediction rates.	Dimensionality problems were not discussed.
6.	Lakshmi	SVM, Adaboost and AdaRank	The system proved that the model with less hidden context have shown greater results when compared to that of with greater hidden context.	Deeper optimization techniques may improve the system more.
7.	Sadiq	DM, DL, and ML-Based Strategy, Keyword Optimization Strategy, URL Scan-Based Method, Blacklisting-Whitelisting Process, and Optical Similarity-Based Strategy	This paper presented a comparative study that focuses on the strategies that over come the existing issues in web phishing and the innovative measures that were taken up before.	Practical study was not undergone. More information regarding optimization techniques is required.
8.	S. -J. Bu	CNN-LSTM	Worked out on real world issues by gathering real time data.	A student-teacher strategy could be implemented further.
9.	A. Odeh	RF, SVM, NB, Adaboost	The article presented a review on the standard ML procedures and discussed both advantages and demerits of the ML techniques. Also explained about the necessary constraints that need to be investigated.	In future, ensemble and DL techniques could be applied.

10.	Yun Lin	RCNN	The desired results have been met after working out over the considered challenges and designed outcome direction.	In future, data collection directly from online could be taken and work out.
11.	I. Saha	Multilayer perceptron, feed forward NN.	The proposed methodology is claimed to be learned not only from the data fed but also could identify the unknown pages efficiently.	In future, more layers in the NN could be added and back propagation could be performed.

From the above literature, the model has identified few important limitations, addressing the solution can improve the performance of the hyper tune model. Traditional CNN approaches use standard number of estimators irrespective of the data available for training. Any learning algorithms needs to get training based on the data and parameters hyper tuning using Meta heuristic approach helps the model to have better performance. This algorithm identifies the parameters which get acquainted to network based on the current instance of epoch and the random samples in the present iteration.

3. PROPOSED METHODOLOGY

The proposed research initially identifies the important features using the pipelined REF, which is combined with Boosting classifier. To design a customized neural network, it has to first identify the optimized values for each estimator using the enhanced Bayesian optimizer

. **Algorithm for Customized Neural Network:**

```
Input: Load the phishing dataset, PD
Output: Evaluation Metrics computation
Begin:
1.  Filling of NAN, Null values and replacing infinity values
2. Splitting the dataset into training and test with traditional ratio values
3. Apply standardization to transform the data
4. Select the important features using REF combined with XGBOOST in pipeline
5. Define independent and dependent attributes
6. Call Optimizer function to find the best values by defining the maximization objective function
7.  Print the optimized estimator values
8. Fit the neural networks with the obtained values
9. Compute the accuracy, recall and precision
End.
```

Algorithm forEnhanced Bayesian Optimizer:

```
Input: Neurons, Activation, Optimizer, Learning Rate, Batch Size, Epochs, Layers, Normalization, Drop Out, Dropout Rate

Output: Optimized Values for all the estimators

Begin:

1. Create a sequential layer
2. Define a set of dictionaries that consists all the possible optimizers like Adam, SGD, RMSProp, and others for both convolution and dense layers
3. Define activation functions
4. Define neurons in the range of 10 to 1000, batch size in the range of 200 to 1000
5. Define 3 dense layers and drop out layers with a threshold greater than 0.5
6. Define a dense output layer with sigmoid activation function
7. Compute the cross validation score by applying stratified folding technique.
End.
```

The algorithm tries to find the values for the estimators described in the table 3.

Table 3: Training Parameters for the neural networks

S.No	Estimators	Description	Best Value Obtained
1	Number of Neurons	The neuron acts as a transferring agent to combine inputs from different layers. In general, it is better to have less than twice of input size	40
2	Activation	It act as a conversion function which manipulates each element from linear to non-linear	Elu
3	Optimizer	It act as objective function to minimize the loss incurred during the training phase	RMSProp
4	Learning_rate	It defines the change of rate which makes	0.546

		the system to adapt to the new environment	
5	Batch_size	It works on the count of samples to be updated to make the model gets acquainted with environment	335
6	Epochs	It defines the number of iterations that a model has to undergo for training purpose	44
7	Number of Layers	It plays a vital role in deciding the dimensionality that can be solved by the model	3
8	Normalization	It solves the problems occurred due to unbiased value raised by high level features and it also reduces the training time exponentially	0.99
9	Drop out	It regularizes the neural network and prevents the problems caused by overfitting	0.43
10	Drop rate	It defines the learning rate for each layer	0.38

The algorithm uses dense layer i.e., a full connected layer as its output because its reduces the dimensionality of the preceding layers to simplify the relation between different layers involved in the network design. The architecture for the detecting the phishing website or not is shown in the figure 4.

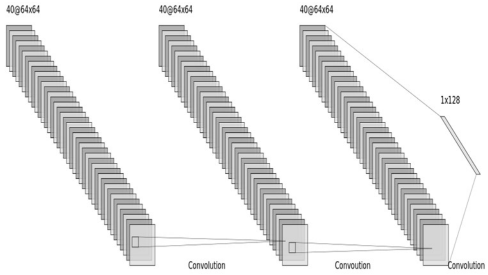


Figure 4: Architecture for the Designing the Model

In the designed architecture, figure 4, initially, the model contains a sequential network with three dense layers and one output layer. Each layer in the networks contains 40 neurons. A learning rate to train the model is statically defined but after epochs it changes. Selection of learning rate is a crucial step because higher the learning rate the model misses the high end features. So, the proposed algorithm has chosen the moderate value. Since, dataset contains plenty of records; each layer in every epoch gets trained on 335 records. In between the dense layers, the model incorporates normalization layer with 0.99 values to scale all the input values before passing it as input to the next layer. Each layer is associated with ELU activation and ADAM optimizer to handle the non-linear properties of the data.

In this model, the last layer gets the input from every neuron of the previous layer and to compute the output, it performs the matrix vector multiplication and produces N-dimensional matrix. This can be mathematically represented as shown in figure 5.

$$\begin{array}{cccc} \begin{pmatrix} 10 & 25 \\ 30 & 70 \end{pmatrix} & * & \begin{pmatrix} 1 & 0 \\ 0.1 & 0.4 \end{pmatrix} & = \begin{pmatrix} 12.5 & 10 \\ 37 & 28 \end{pmatrix} \\ \textbf{Feature Matrix (X)} & & \textbf{Weight Matrix (W)} & \textbf{Dense Layer O/P} \end{array}$$

Figure 5: Output Computation at Dense Layer

The main aim of the activation function is to produce the output with non- linear combinations, because linear combinations fail to recognize complex structures like image, audio, and video and the implementation of sigmoid can handle even the deep neural network. The model wants to find the probability for the classification, which has to produce the output as either 0 or 1.

Deciding the number of neurons in the hidden layers is crucial element for any neural network, the Bayesian optimizer computes the requirement using the equation (1)

$$h(n) = \frac{s(n)}{\propto * (i(n) * o(n))} \text{ - (1)}$$

where,

h(n) denotes number of neurons in hidden layer

s(n): number of training samples

i(n): number of neurons in input layer

o(n): number of neurons in output layer

α: scaling factor

To solve the problem of gradient descent problem in neural networks, different flavours of linear units in terms of activation function exists like ReLu, SELU, GELU, and others but the proposed research has identified that "ELU (Exponential Linear Unit)" performs well for the detection of phishing attacks, the computation of which is illustrated in equation (2)

if(input)= input, if input>0

$$= \alpha*(e^{input} - 1), otherwise \qquad \text{eq-2}$$

The main advantage of this approach is it tries avoiding the production of zero's and making the neurons activated either be producing positive or negative values. It biases the weights and tries to explore the graph in a single direction with a standard geometrical path.

The proposed model has identified "RMSProp" as the best optimizer to design this neural network, which is treated as first order derivative and its gradients, has the power to solve complex derivations by considering their average values to produce output and always ensures that the learning rate is adaptive so that it can fasten the training process. The equation for the optimizer is shown in the equation (3)

$$Weight = Weight - \propto * \frac{d(Weight)}{\sqrt{gradient\ of\ weight}} - (3)$$

To stabilize the neural network, the system implements batch normalization by finding the mean and variance of every iteration data that is defined as the batch then it finds the normalization vector for activation function as shown in equation (4)

$$Norm(h_i) = \frac{Norm(i) - mean\ value\ of\ the\ layer}{\sqrt[2]{standard\ deviation}} \text{ - (4)}$$

The batch normalization process helps the system to pre-process the huge amount of data in few steps to bring to the common scale value. The process of normalization continues till the system completes the training process. The overall designing process for detection of phishing attacks using optimization parameters is illustrated in figure 6,

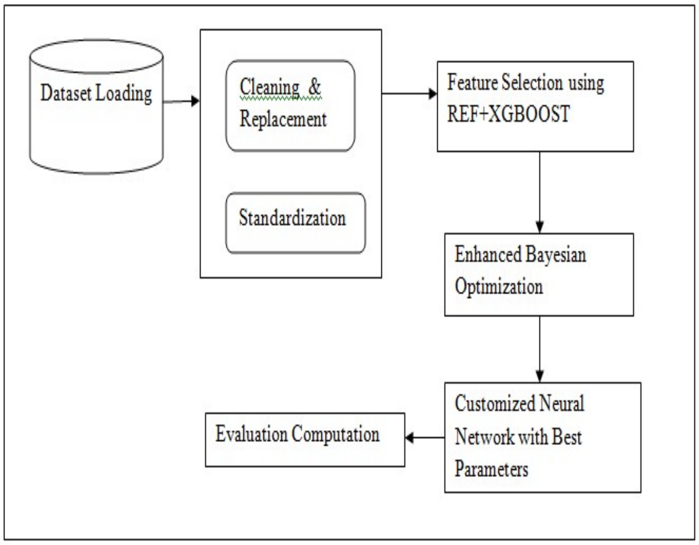
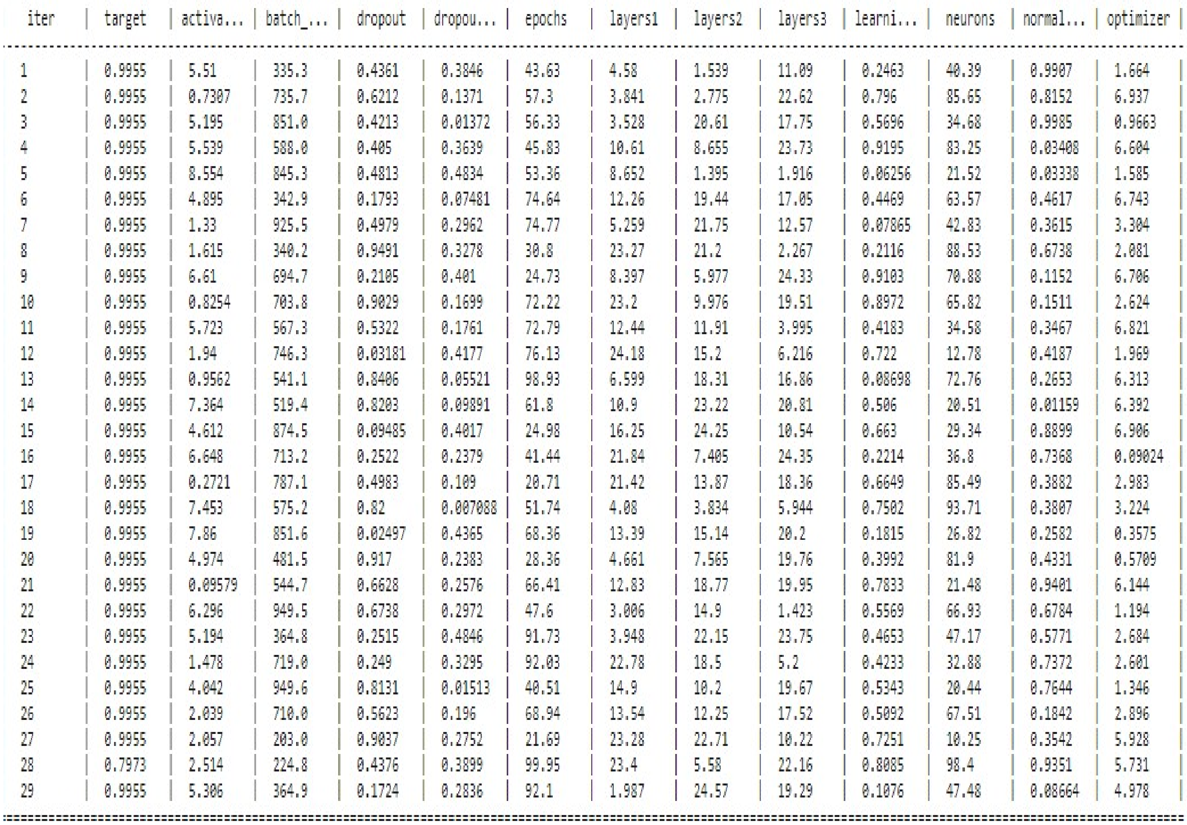

Dataset Loading
Cleaning & Replacement
Standardization
Feature Selection using REF+XGBOOST
Enhanced Bayesian Optimization
Customized Neural Network with Best Parameters
Evaluation Computation

Figure 6: Overview of the Enhanced Bayesian Optimized Neural Network

4. RESULTS AND DISCUSSION

Figure 7 discusses about finding the best parameters by the enhanced Bayesian optimizer to find the best values for all the possible estimators.



```
  iter   | target | activa... | batch_... | dropout | dropou... | epochs | layers1 | layers2 | layers3 | learni... | neurons | normal... | optimizer |
-------------------------------------------------------------------------------------------------------------------------------------------------
  1      | 0.9955 | 5.51      | 335.3     | 0.4361  | 0.3846    | 43.63  | 4.58    | 1.539   | 11.09   | 0.2463    | 40.39   | 0.9907    | 1.664     |
  2      | 0.9955 | 0.7307    | 735.7     | 0.6212  | 0.1371    | 57.3   | 3.841   | 2.775   | 22.62   | 0.796     | 85.65   | 0.8152    | 6.937     |
  3      | 0.9955 | 5.195     | 851.0     | 0.4213  | 0.01372   | 56.33  | 3.528   | 20.61   | 17.75   | 0.5696    | 34.68   | 0.9985    | 0.9663    |
  4      | 0.9955 | 5.539     | 588.0     | 0.405   | 0.3639    | 45.83  | 10.61   | 8.655   | 23.73   | 0.9195    | 83.25   | 0.03408   | 6.604     |
  5      | 0.9955 | 8.554     | 845.3     | 0.4813  | 0.4834    | 53.36  | 8.652   | 1.395   | 1.916   | 0.06256   | 21.52   | 0.03338   | 1.585     |
  6      | 0.9955 | 4.895     | 342.9     | 0.1793  | 0.07481   | 74.64  | 12.26   | 19.44   | 17.05   | 0.4469    | 63.57   | 0.4617    | 6.743     |
  7      | 0.9955 | 1.33      | 925.5     | 0.4979  | 0.2962    | 74.77  | 5.259   | 21.75   | 12.57   | 0.07865   | 42.83   | 0.3615    | 3.304     |
  8      | 0.9955 | 1.615     | 340.2     | 0.9491  | 0.3278    | 30.8   | 23.27   | 21.2    | 2.267   | 0.2116    | 88.53   | 0.6738    | 2.081     |
  9      | 0.9955 | 6.61      | 694.7     | 0.2105  | 0.401     | 24.73  | 8.397   | 5.977   | 24.33   | 0.9103    | 70.88   | 0.1152    | 6.706     |
  10     | 0.9955 | 0.8254    | 703.8     | 0.9029  | 0.1699    | 72.22  | 23.2    | 9.976   | 19.51   | 0.8972    | 65.82   | 0.1511    | 2.624     |
  11     | 0.9955 | 5.723     | 567.3     | 0.5322  | 0.1761    | 72.79  | 12.44   | 11.91   | 3.995   | 0.4183    | 34.58   | 0.3467    | 6.821     |
  12     | 0.9955 | 1.94      | 746.3     | 0.03181 | 0.4177    | 76.13  | 24.18   | 15.2    | 6.216   | 0.722     | 12.78   | 0.4187    | 1.969     |
  13     | 0.9955 | 0.9562    | 541.1     | 0.8406  | 0.05521   | 98.93  | 6.599   | 18.31   | 16.86   | 0.08698   | 72.76   | 0.2653    | 6.313     |
  14     | 0.9955 | 7.364     | 519.4     | 0.8203  | 0.09891   | 61.8   | 10.9    | 23.22   | 20.81   | 0.506     | 20.51   | 0.01159   | 6.392     |
  15     | 0.9955 | 4.612     | 874.5     | 0.09485 | 0.4017    | 24.98  | 16.25   | 24.25   | 10.54   | 0.663     | 29.34   | 0.8899    | 6.906     |
  16     | 0.9955 | 6.648     | 713.2     | 0.2522  | 0.2379    | 41.44  | 21.84   | 7.405   | 24.35   | 0.2214    | 36.8    | 0.7368    | 0.09024   |
  17     | 0.9955 | 0.2721    | 787.1     | 0.4983  | 0.109     | 20.71  | 21.42   | 13.87   | 18.36   | 0.6649    | 85.49   | 0.3882    | 2.983     |
  18     | 0.9955 | 7.453     | 575.2     | 0.82    | 0.007088  | 51.74  | 4.08    | 3.834   | 5.944   | 0.7502    | 93.71   | 0.3807    | 3.224     |
  19     | 0.9955 | 7.86      | 851.6     | 0.02497 | 0.4365    | 68.36  | 13.39   | 15.14   | 20.2    | 0.1815    | 26.82   | 0.2582    | 0.3575    |
  20     | 0.9955 | 4.974     | 481.5     | 0.917   | 0.2383    | 28.36  | 4.661   | 7.565   | 19.76   | 0.3992    | 81.9    | 0.4331    | 0.5709    |
  21     | 0.9955 | 0.09579   | 544.7     | 0.6628  | 0.2576    | 66.41  | 12.83   | 18.77   | 19.95   | 0.7833    | 21.48   | 0.9401    | 6.144     |
  22     | 0.9955 | 6.296     | 949.5     | 0.6738  | 0.2972    | 47.6   | 3.006   | 14.9    | 1.423   | 0.5569    | 66.93   | 0.6784    | 1.194     |
  23     | 0.9955 | 5.194     | 364.8     | 0.2515  | 0.4846    | 91.73  | 3.948   | 22.15   | 23.75   | 0.4653    | 47.17   | 0.5771    | 2.684     |
  24     | 0.9955 | 1.478     | 719.0     | 0.249   | 0.3295    | 92.03  | 22.78   | 18.5    | 5.2     | 0.4233    | 32.88   | 0.7372    | 2.601     |
  25     | 0.9955 | 4.042     | 949.6     | 0.8131  | 0.01513   | 40.51  | 14.9    | 10.2    | 19.67   | 0.5343    | 20.44   | 0.7644    | 1.346     |
  26     | 0.9955 | 2.039     | 710.0     | 0.5623  | 0.196     | 68.94  | 13.54   | 12.25   | 17.52   | 0.5092    | 67.51   | 0.1842    | 2.896     |
  27     | 0.9955 | 2.057     | 203.0     | 0.9037  | 0.2752    | 21.69  | 23.28   | 22.71   | 10.22   | 0.7251    | 10.25   | 0.3542    | 5.928     |
  28     | 0.7973 | 2.514     | 224.8     | 0.4376  | 0.3899    | 99.95  | 23.4    | 5.58    | 22.16   | 0.8085    | 98.4    | 0.9351    | 5.731     |
  29     | 0.9955 | 5.306     | 364.9     | 0.1724  | 0.2836    | 92.1    | 1.987   | 24.57   | 19.29   | 0.1076    | 47.48   | 0.08664   | 4.978     |
=================================================================================================================================================
```

Figure 7: Training Process to find best parameter values

Table 4 compares the previous works carried out by various researchers and proposed research and proves its calibre in terms of accuracy metric.

Table 4: Proposed Versus Existing Research Works

S.No	Algorithm	Accuracy	Recall	Precision
1	DT, kNN, RF ensemble [2]	97.65	90	89
2	DNN- LSTM and DNN – BiLSTM [4]	98.1	95.3	94.2
3	NB, RNN [5]	92.26	89.1	90.3
4	SVM, Adaboost and AdaRank [6]	90	90	90
5	Multilayer perceptron, feed forward NN. [11]	95	93	93
6	Proposed	99.5	99	99

From figure 8, it is clearly evident that the proposed system has not only improved in terms of accuracy but also in terms of recall and precision. From the above figure, the proposed system also found that machine learning techniques has less performance than deep learning approaches. Table 5 compares the traditional deep learning, cross fold validated deep learning and optimized deep learning results to prove its efficiency by evaluating various parameters.

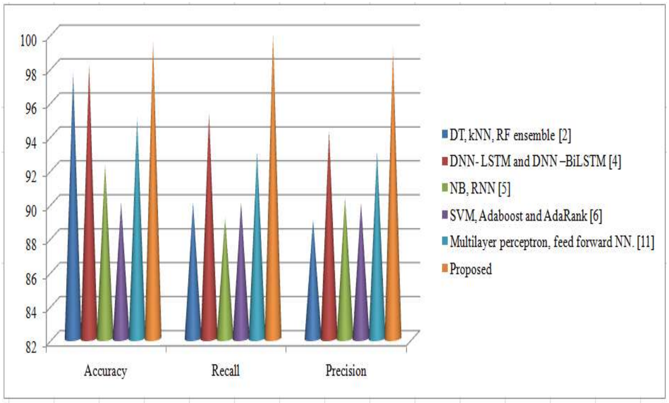


Figure 8: Comparison of Existing and proposed works

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5: Efficiency in terms of evaluation metrics

Algorithm Name	Accuracy	Recall	Precision
ANN	98.7	100	99
Cross validated	98.6	100	99
Parameterized	99.5	99	99

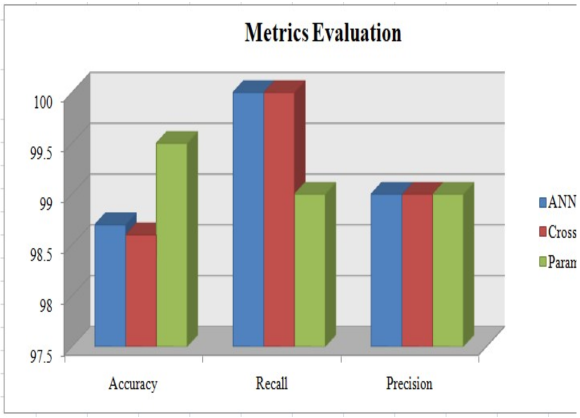


Figure 9: Deep Learning Algorithms Evaluation Analysis

The figure 9 represents the comparison of deep learning algorithms based on accuracy, recall and precision. It is observed that basically, the accuracy in proposed research is improved over the traditional approaches. In terms of recall and precision also, it works in parallel to naïve approaches.

5. CONCLUSION

The major focus of this proposed research is to prove the efficiency of the model by customizing the Bayesian optimizer algorithm to find the best parameters by finding the number of layers, neurons and other metrics based on the data trained during the system designing process. The research has identified that machine learning approaches are good but are not efficient. So, the proposed system to improve the evaluation process it implemented a customized neural network which has increased the learning rate, number of epochs, drop rate than the existing works. Since the learning rate is moderate, the adaptability of the system increases during the training phase. The previous work with hyper tuned machine learning algorithm, the research has got 99.2% accuracy where as with the EBONN algorithm, the research has achieved 99.7% accuracy. The proposed algorithm efficiently identifies the accurate estimator values of necessary parameters of neural network by identifying the required number of layers and neurons based on the data available. In future work, the activation functions can be chosen based on the genetic algorithm by identifying the dying point in the network rather than choosing the existing functions. An integrated combinations can improve the normalization vectors regarding non-linearity functions.

REFERENCES

[1] Basit, A., Zafar, M., Liu, X. et al. A comprehensive survey of AI-enabled phishing attacks detection techniques. TelecommunSyst 76, 139–154 (2021).https://doi.org/10.1007/s11235-020-00733-2.

[2] M. Toğaçar , "Detection of Phishing Attacks on Websites with Lasso Regression, Minimum Redundancy Maximum Relevance Method, Machine Learning Methods, and Deep Learning Model", Turkish Journal of Science and Technology, vol. 16, no. 2, pp. 231-243, Sep. 2021.

[3] Priya, S., Selvakumar, S. &Velusamy, R.L. Evidential theoretic deep radial and probabilistic neural ensemble approach for detecting phishing attacks. J Ambient Intell Human Comput (2021).https://doi.org/10.1007/s12652-021-03405-4.

[4] Ozcan, A., Catal, C., Donmez, E. et al. A hybrid DNN–LSTM model for detecting phishing URLs.Neural Comput&Applic (2021).https://doi.org/10.1007/s00521-021-06401-z.

[5] Rani Shinde1, Dr. GitanjaliShinde. et al. Illegitimate Websites Detection Using Deep Learning Framework (2021). DOI: https://doi.org/10.51319/2456-0774.2021.7.0025

[6] Lakshmi, L., Reddy, M. P., Santhaiah, C., & Reddy, U. J. (2021). Smart Phishing Detection in Web Pages using Supervised Deep Learning Classification and Optimization Technique ADAM. Wireless Personal Communications, 118(4), 3549–3564.doi:10.1007/s11277-021-08196-7.

[7] Sadiq, A., Anwar, M., Butt, R. A., Masud, F., Shahzad, M. K., Naseem, S., &Younas, M. (2021). A review of phishing attacks and countermeasures for internet of things-based smart business applications in industry 4.0.In Human Behavior and Emerging Technologies.Wiley. https://doi.org/10.1002/hbe2.301

[8]S. -J. Bu and S. -B. Cho, "Integrating Deep Learning with First-Order Logic Programmed Constraints for Zero-Day Phishing Attack Detection," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 2685-2689, doi: 10.1109/ICASSP39728.2021.9414850.

[9] A. Odeh, I. Keshta and E. Abdelfattah, "Machine LearningTechniquesfor Detection of Website Phishing: A Review for Promises and Challenges," 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0813-0818, doi: 10.1109/CCWC51732.2021.9375997.

[10] Lin, Yun & Liu, Ruofan&Divakaran, Dinil Mon & Ng, Jun & Chan, Qing & Lu, Yiwen& Si, Yuxuan& Zhang, Fan & Dong, Jin. (2021). Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages (USENIX Security 2021).

[11] I. Saha, D. Sarma, R. J. Chakma, M. N. Alam, A. Sultana and S. Hossain, "Phishing Attacks Detection using Deep Learning Approach," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 1180-1185,

doi: 10.1109/ICSSIT48917.2020.9214132.