

Investigative Study of the Effect of Various Activation Functions with Stacked Autoencoder for Dimension Reduction of NIDS using SVM

Nirmalajyothi Narisetty¹, Gangadhara Rao Kancherla²
Basaveswararao Bobba³
Dept. of CSE, Acharya Nagarjuna University
Guntur, 522510, India

K.Swathi⁴
Dept. of CSE
NRI Institute of Technology
Agiripalli, India

Abstract—Deep learning is one of the most remarkable artificial intelligence trends. It remains behind numerous recent achievements in various domains, such as speech processing, and computer vision, to mention a few. Likewise, these achievements have sparked great attention in utilizing deep learning for dimension reduction. It is known that the deep learning algorithms built on neural networks contain number of hidden layers, activation function and optimizer, which make the computation of deep neural network challenging and, sometimes, complex. The reason for this complexity is that obtaining an outstanding and consistent result from such deep architecture requires identifying number of hidden layers and suitable activation function for dimension reduction. To investigate the aforementioned issues linear and non-linear activation functions are chosen for dimension reduction using Stacked Autoencoder (SAE) when applied to Network Intrusion Detection Systems (NIDS). To conduct experiments for this study various activation functions like linear, Leaky ReLU, ELU, Tanh, sigmoid and softplus have been identified for the hidden and output layers. Adam optimizer and Mean Square Error loss functions are adopted for optimizing the learning process. The SVM-RBF classifier is applied to assess the classification accuracies of these activation functions by using CICIDS2017 dataset because it contains contemporary attacks on cloud environment. The performance metrics such as accuracy, precision, recall and F-measure are evaluated along with these classification time is being considered as an important metric. Finally it is concluded that ELU is performed with low computational overhead with negligible difference of accuracy that is 97.33% when compared to other activation functions.

Keywords—Auto-encoder; cloud computing; dimension reduction; intrusion detection system; machine learning

I. INTRODUCTION

The Cloud services availability to the individuals, organizations, and Governments connected through web-enabled devices across the world on pay-as-you-go premise [1] have become very common. The security and privacy problems get magnified as new type of attacks when internet environment migrate to Cloud [2]. Among these types of malicious activities Distributed Denial of Service (DDoS) attacks are easily invoked by the attackers basically with malicious intent of denying the Cloud services.

This type of attacks causes the interruption of cloud services to legitimate users by inordinate resource consumption which would automatically results in Service Level Agreement (SLA) violation. Most of the Cloud services are inherently elastic so the DDoS attacks are damaging the cloud service provider (CSP) economically but not its physical system or server assets [3]. This phenomenon is known as Economic Denial of Sustainability (EDoS) EDoS attack.

Due to increase in migration to the cloud the security and privacy problems are also increased in cloud environment with new types of malicious activity penetrated by professionals with cutting edge technologies with malicious intent. Over the last three decades the problems related to security and privacy are major research problems and addressed by several researchers with the evaluation of Network Intrusion Detection Systems (NIDS). It is a necessity to improve the NIDS to mitigate the new type of attacks on cloud because several users are getting migrated to cloud environment. It is a necessary to improve the NIDS to mitigate the attacks on cloud environment and this problem is addressed by several researchers. They have identified significant measures to detect and mitigate such types of attacks using statistical, Machine Learning (ML) techniques and knowledge based approaches. The NIDS needs to be more robust to increase the users trust in adoption of cloud computing in future. Therefore Network traffic analysis of cloud is necessary to identify the patterns in order to discriminate malicious users and legitimate users. ML approaches offer great strength and diversity in research for anomaly detection in NIDS using classification task [4].

Lot of significant research work is going on over the past two decades on dimension reduction using Wrapper methods, Filter based Approaches, etc. to address the curse of dimensionality [5]. One of the advantages of NIDS is the availability of huge collection of network data related to cloud environment on which machine learning algorithms can be applied to detect attacks. Such a complex and huge data may disgrace the performance metrics of classifiers [6]. Dimensionality reduction or feature learning is one of the stages in classification which helps to extract the relevant information from the original data to reduce the computational time without compromising the other performance metrics.

Recently Deep learning methods from the family of Machine Learning approaches are successfully applied to extract good feature representation automatically [7]. Due to its capability for extracting valuable and useful information from large data yields better classification process and lower complexity.

The chosen activation function plays an important role in deep learning models to improve the accuracy rate and reduce computational complexity [8]. Activation function is one of the principal factors which will affect the performance of the neural networks [9]. Activation functions are basically divided into linear or non-linear relying upon the function it represents. Activation functions are leveraged to transfer and control the outputs of neural networks, across various domains from object recognition and classification.

This research work proposes a comprehensive investigative study intended to identify suitable activation function of SAE. By implementing this activation function an attempt is made to extract an optimized feature subset. The SAE extracts key features from the CICIDS2017 which is exposed to lot of vulnerabilities as it is on Internet. Then a SVM classifier stage is used to classify the attacks and distinguish between normal and anomalous traffic.

After examining the contemporary studies the SAE model is identified as one of the best model for dimension reduction of various types of bigdata analytical content like image, text and network intrusion detection. There is a need to identify the proper activation function that will be more effective in the process of dimension reduction using SAE by satisfying the criteria like high classification accuracy and minimum classification time.

In view of the above the problem is selected to identify suitable activation function of SAE for dimension reduction and evaluate the classification accuracies. To achieve this objective the study is carried out with following contributions.

- Sharp decisions are needed within a stipulated time which plays an important role for NIDS. So to suggest a novel framework for better NIDS with suitable activation function and classifier.
- To compare the different activation functions of SAE using SVM classifier with RBF kernel.
- To evaluate the performance metrics and computational time with adoption of CICIDS2017 dataset.
- Finally identified the effective activation function based on experimental results.

The rest of this paper is organized as follows: the related previous work is outlined in Section 2 and Section 3 is explains the description of the CICIDS2017 dataset. The methodology and experimental setup of the proposed model is depicted in Section 4. The results and discussions are presented in Section 5. Finally conclusions and future scope of this work are given in Section 6.

II. LITERATURE REVIEW

An IDS has been studied for the last two decades using various machine learning approaches. This section will discuss some of the approaches proposed by researchers to analyze the effect of activation functions for IDS as well as for feature selection.

In [10], Investigated the performance of different types of rectified activation functions using Convolutional neural network. The standard rectified activation functions of rectified linear unit (ReLU), leaky rectified linear unit (Leaky ReLU), parametric rectified linear unit (PReLU) and a new randomized leaky rectified linear unit (RReLU) are evaluated with three different dataset NDSB, CIFAR-10 and CIFAR-100. Observations are discussed based on the exploratory results. However, this study is limited to different types of rectified activation functions and not pertaining to large datasets.

In [11], studied the effect of Activation functions on classification accuracy using Deep Artificial Neural Networks (ANN). The ANN is used to classify real multi-spectral Landsat 7 satellite images and therefore the accuracy of the classification was evaluated with twelve different activation functions. The accuracy of classifier can be improved by selecting good activation function. However, the activation functions sigmoid and bipolar activation functions are recommended to specific field of remote sensing.

It is Observed and stated in [12] that activation functions play pivotal role in understanding neural network. "DBN suffers from vanishing gradient problem due to the saturation characteristic of activation function. Therefore, the selection of activation function in DBN is critical to reduce the network complexity and to improve the performance of pattern recognition". DBN based classification with different types of activation functions like Sigmoid, Hyperbolic Tangent, MSAF, ReLU and LReLU can be used to examine their performance with MNIST dataset. Besides that, the randomization of training samples would significantly improve the performance of DBN. The experimental results showed that hyperbolic tangent activation function achieved the lowest error rate which is 1.99% on MNIST handwritten digit dataset.

Kunang, Y. N et al. in [13] presented a deep learning based dimension reduction approach in the first step using stacked autoencoder. The input neurons 120 were reduced to 8 neurons with seven activation functions i.e., linear, sigmoid, ReLU, SoftMax, Softplus, Softsign and tanh and couple of loss functions like Mean Squared error and cross entropy to seek out the suitable activation function. In the second step reduced data is fed as input to supervised classifier SVM with RBF kernel to evaluate the output of first step. Based on the results it's shown that the ReLU and linear activation functions with cross entropy yields better results over the other combinations.

The authors in [14] compared non-linear activation functions alternative to sigmoid in deep neural networks on Image Dataset, and also different weight initialization methods Gaussian distribution, uniform distribution, learning rate from 0.01 to 0.2, batch size 50 and 100 epochs and effect of hidden layers were tested. Based on the experimental results it was noticed that the accuracies of ReLU and its variants are higher than the sigmoid activation function and the learning rate is faster in ELU and SELU compared to the ReLU and Leaky ReLU. However, this study did not pay attention to computational complexity.

In [15] fast activation function adaptive Linear Function (ALF) is proposed for anomaly detection to increase the speed and accuracy of the deep leaning structure for real-time applications. Deep Belief Network with new activation function ALF and Sigmoid, Tanh, and ReLU are used for classification on four datasets namely NSL KDD, Kyoto, KDDCUP'99 and CSIC 2010. The experiments were conducted with a combination of each dataset using all four activation functions. Exploratory results have shown that learning structure using the fast ALF activation function outperforms the state-of-the-art network Stacked Sparse AutoEncoder Based Extreme Learning Machine (SSAELM) in accuracy and convergence time. However, the proposed approach needs to be evaluated on contemporary datasets.

Another research work [16] built IDS model with a blend of two models, autoencoder and deep artificial neural network. In the first stage Stacked autoencoder based dimension reduction with sigmoid activation function in input/hidden layer/output layer using 4 hidden layers and reconstruction error is calculated with mean squared loss function performed. In the next stage deep artificial neural network model was built utilizing different activation functions like hard sigmoid, relu, sigmoid, softplus and tanh to further classify the network traffic. Conclusions are drawn based on F1-score which is best with relu activation function in comparison to other activation functions. However, this study is limited to binary classification and need to, be tested on multi-class classification.

Feng, J., & Lu, S in [9] studied the characteristics of linear and non-linear activation functions Sigmoid, Tanh, ReLU, LeakyReLU, PReLU, RReLU, and ELU are compared. Activation functions advantages and disadvantages in Artificial Neural Network are also discussed and concluded that choosing of suitable activation function depends on our aim and network structure.

A Novel method was proposed by [17] to test SVM classifier with different kernels and Deep Convolutional Neural Network (DCNN) method for classification with different activation functions ReLU, Sigmoid, SoftMax, and Tanh. Before performing the intrusion detection chi-square based

feature selection was performed to lessen the size of NSL-KDD dataset. The experiments were conducted with different activation functions by fixing one function from the above said four activation functions in input and hidden layers. Then the remaining are used at output layer respectively for each experiment. DCNN yields better results with sigmoid as output layer activation function with any of the activation function in input/hidden layers. The performance of SVM and DCNN are analyzed and concluded that the DCNN performed well compared to SVM classifier in terms of accuracy. This study focused on binary classification.

Le, T. T. H. et al. explored six non-linear Activation functions i.e., Softplus, ReLU, Tanh, Sigmoid, ELU and Leaky ReLU in [18] and additionally the impact of those activation functions was analyzed with Recurrent Neural Network (RNN) model to find the best activation function for intrusion detection. The KDD cup dataset was employed to conduct experiments. Among the employed activation functions Leaky ReLU the best results in terms of performance metrics accuracy, precision, recall and False Alarm Rate (FAR). The study is associated with performance metrics but does not concentrate on time complexity.

Most of the above mentioned studies focused mainly on comparing the various activation functions in evaluation of classifying the performance of neural network models and dimension reduction carried out with autoencoders. They considered the image datasets. In my knowledge and opinion very few studies are carried out by the researchers on network intrusion detection as well as autoencoders being used for dimension reduction on outdated datasets like KDD Cup '99 and NSL-KDD etc. To fulfill these research gaps the present research is carried out with adoption of CICIDS2017 dataset as it contains modern type of attacks and generated on cloud environment.

III. DESCRIPTION OF CICIDS2017 DATASET

The CICIDS2017 dataset generated by [19] is chosen to train the model, which contains benign and the up-to-date common attacks, which resembles the real world data. It includes the network traffic from 09:00 on Monday, July 3rd and continuously ran for an exact duration of 5 days, ending at 17:00 on Friday July 7th. Network data is extracted using CIC Flow Meter, 2017 with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files). The dataset is publicly available in PCAP files¹. In the current study only Wednesday data is considered for intrusion detection. It consists of 6,92,703 instances and 85 feature columns including a label with 6 classes such as Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed. The distribution of records label wise is shown in below Table I.

The feature names and the Attacks distribution of the CICIDS2017 dataset are available in [20].

TABLE I. LABEL WISE DISTRIBUTION OF CICIDS2017 DATASET

Category	Class	Number of Records
Anomaly	DoS GoldenEye	10,293
	DoS Hulk	231,073
	DoS Slowhttptest	5,499
	DoS slowloris	5,796
	Heartbleed	11
Total Anomaly data	---	252,672
Normal	Benign	440,031
Total		692,703

IV. METHODOLOGY

A novel framework is developed for analyzing the performance of various activation functions of SAE for dimension reduction. For evaluation of the performance of these dimension reduction models SVM-RBF classifier is chosen. For this purpose CICIDS2017 dataset is used for conducting experiments. This methodology consists of three phases: A) Data pre-processing, B) Dimension Reduction, and C) SVM-RBF classification. The flow of the proposed framework is depicted in Fig. 1.

A. Data Preprocessings

In the first phase data cleaning and normalization operations are carried out.

Data cleaning: It involves finding the null records as Machine Learning algorithms cannot build a model or test them. Since the percentage of null records associated with each class is small so these records are deleted from the dataset.

Based on the statistical measures of the dataset given in Table II, such as maximum, minimum, standard deviation of each feature it is observed that all the values of attributes Bwd

PSH Flags, Fwd URG Flags, Bwd URG Flags, CWE Flag Count, Fwd Avg Bytes, Fwd Avg Packets, Fwd Avg Bulk Rate, Bwd Avg Bytes, Bwd Avg Packets, Bwd Avg Bulk Rate are null in the dataset. These columns are also deleted from dataset because they potentially don't have any contribution towards classification.

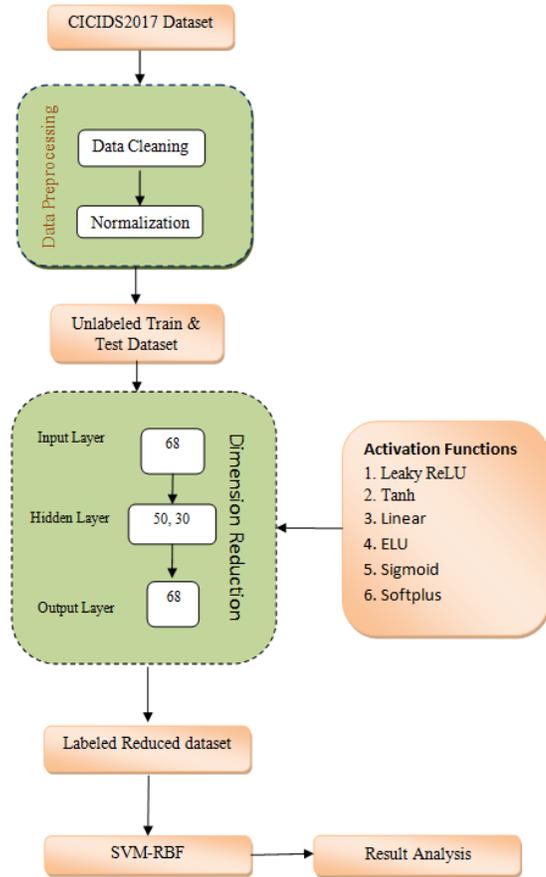


Fig. 1. Novel Framework for Comparison of Activation Functions for NIDS.

TABLE II. STATISTICAL MEASURES OF FEATURES WITH ZERO VALUES

	Bwd PSH Flags	Fwd URG Flags	Bwd URG Flags	CWE Flag Count	Fwd Avg Bytes/Bulk	Fwd Avg Packets/Bulk	Fwd Avg Bulk Rate	Bwd Avg Bytes/Bulk	Bwd Avg Packets/Bulk	Bwd Avg Bulk Rate
count	692703	692703	692703	692703	692703	692703	692703	692703	692703	692703
mean	0	0	0	0	0	0	0	0	0	0
Std	0	0	0	0	0	0	0	0	0	0
min	0	0	0	0	0	0	0	0	0	0
25%	0	0	0	0	0	0	0	0	0	0
50%	0	0	0	0	0	0	0	0	0	0
75%	0	0	0	0	0	0	0	0	0	0
max	0	0	0	0	0	0	0	0	0	0

TABLE III. LABEL WISE DISTRIBUTION OF CICIDS2017 DATASET AFTER DATA CLEANING

Category	Class	Number of Records
Anomaly	DoS GoldenEye	10293
	DoS Hulk	2,30,124
	DoS Slowhttptest	5,499
	DoS slowloris	5,796
	Heartbleed	11
Total Anomaly data	---	2,51,723
Normal	Benign	4,39,972
Total	---	6,91,695

Additionally two features Byte/s and Flow Packet/s have infinity/NaN values in very few records which are replaced with zeros to avoid difficulties in applying the ML algorithms. After performing data cleaning operation the distribution of the remaining dataset is given Table III.

Normalization: ¹By nature the features of the dataset are quantitative. Several features of the dataset lie in a wide range mostly between the highest possible value and lowest possible value.

The features with higher values may dominate the features with low values. To avoid this, features to be normalized to eliminate such dominance before applying ML algorithms.

Normalization is a scaling technique in which values are transformed and rescaled so that they fall in specific range [0, 1]. In this paper features are scaled using min-max normalization. The feature f values with a range between f_{min} and f_{max} , then the normalization is defined by the equation of $f_{nor} = (f_i - f_{min}) / (f_{max} - f_{min})$, Where f_{nor} is a normalized value of the i th value of feature f . An illustrative example of normalization for one record is given in Fig. 2.

B. Dimension Reduction using SAE

This section explores the impact of the different activation functions of SAE with specified hyper parameters as given below for feature dimensionality reduction.

The performance of NIDS using neural networks for dimension reduction as well as classification is dependent on three criteria. They are i) to identify suitable activation function, ii) the number of hidden layers to be used and iii) to

adjust the proper weights for minimizing the loss between input and output. Using SAE for dimension reduction the chosen of activation function is a crucial problem.

Among the three the first one plays a crucial role for optimizing the feature subset. So this study considers to find the impact of the first criterion and the remaining are fixed at number of hidden layers are 3 and Mean Square Error as a loss function. The main focus of this study is to compare various prominent activation functions that are used in SAE for NIDS and evaluate them through the SVM classifier. Proper activation function also affects the classification time along with classification accuracy [11]. For this purpose six activation functions are chosen. The following Fig. 3 depicts the typical structure of the SAE which consists of input and output layers with three hidden layers, the same is being considered for conducting experiments.

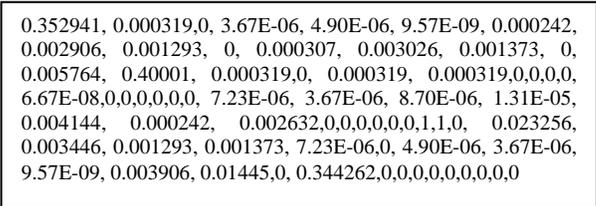


Fig. 2. Resulting One Sample Record after Normalization.

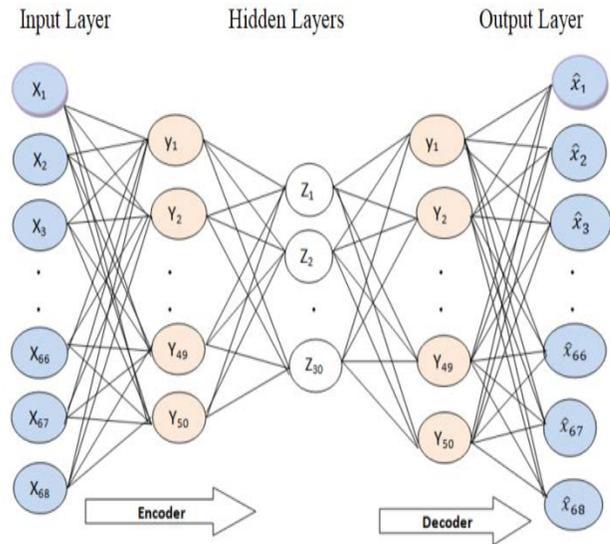


Fig. 3. Typical Structure of Stacked Autoencoder.

¹ <http://www.unb.ca/cic/datasets/IDS2017.html>.

The below SAE has 68 neurons in both input and output layers which are equal to the features of normalized train and test dataset. The numbers of neurons in three hidden layers are 50, 30 and 50 respectively. The role of activation function in neural network is to facilitate the transfer of data from input layer neurons to output layer neurons [18]. The chosen six activation functions are being applied.

Several different kinds of optimizers are used by different researchers even though their common purpose is minimizing the loss in learning process. Among those Adaptive Moment Estimation (Adam) is chosen for this study [21]. The following function is adopted for reconstruction error computation [22]. The mean square error function is calculated for the samples of x_i and reconstructed samples \hat{x}_i as shown in equation 1. To find the optimal structure of SAE for dimension reduction with afore mentioned optimizer and loss function used. The unlabeled normalized train and test datasets of CICIDS2017 are taken as input datasets, because SAE is an unsupervised dimension reduction approach.

$$d(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^N \|x_i - \hat{x}_i\|^2, \quad (1)$$

where N is the number of input samples.

The following algorithms takes various hyper parameters weight vector W, bias vector, batch_size, epochs, number of layers (L), learning rate α and neurons_structure in each layer are assigned to initial values. The training procedure of unsupervised SAE model for dimension reduction is given in Algorithm1 and Algorithm2. The input to Algorithm1 is number of SAE structures, number of hidden layers in each structure. Step3 inputs the number of neurons in each hidden layer. Followed by in step4 compute_MSE () is invoked for each structure to setup the corresponding SAE architecture and iterated for all epochs with given activation function then returns the MSE error which is stored in loss. Input to Algorithm2 are train and test datasets $\{x_1, x_2, \dots, x_{68}\}$ where sample $X \in \mathbb{R}$ where, structures of SAE and the number of neurons in each structures. For each layer l the parameters are initialized to zero. In each iteration the hidden representation vector of layer l is computed based on the previous layer. After that the loss is calculated equation (4) in step 2.2.2, Next step is to update θ' based on loss and completes all iterations similarly. Finally mean of all the epochs MSE loss is calculated and returned. Once the MSE losses of different structures are calculated then the minimum loss is identified and the corresponding structure is selected as optimized SAE structure for dimension reduction. All the above steps are repeated for each activation function.

Algorithm1: The training procedure to identify optimal SAE structure for dimension reduction is given in Algorithm1

Algorithm1: Optimized SAE model for dimension reduction

Input: S // number of SAE structures
L // number of layers in structure
neurons_structure [] [] // array of SAE structures

Output: Optimized SAE structure

Step 1: // Initialization
S=0, neurons_structure [] [] <- 0, loss []

Step 2: read S //Total Number of Structures

Step 3: for I in 1 to S do // for each structure I

Step 3.1: read L [I] // Number of Layers in the structure

Step 3.2: for J in 1 to L [I] do

Step 3.2.1: read N // number of neurons in Layer J of structure I

Step 3.2.2: store neurons_structure [I] [J] =N

end for

end for

Step 4: for s in 1 to S do

loss[s] = compute_MSE(s, neurons_structure [I], L [I], bias[I])

end for

Step 5: bestvalidation_loss= loss [1]

Step 5.1 for k in 2 to S

Step 5.1.1 if (bestvalidation_loss > loss[k])

then

bestvalidation_loss=loss[k]

location = k

end if

end for

Step 6: optimal structure= neurons_structure [location] []

As a part of initiation towards optimization various structures of neural networks are used to reduce the MSE as per equation7. In the lines of [13] several experiments were conducted for the number of features and the corresponding MSEs are listed in Table IV. It is observed from the above table.

Algorithm2: The training procedure of unsupervised SAE model for dimension reduction is given in Algorithm2

```

Algorithm2: Compute_MSE (s, neurons_structure [ ], L) // Training SAE and calculate MSE

Input: Unlabeled normalized train dataset (UNTR) and test dataset (UNTS), UNTR & UNTS  $\subseteq \mathbb{R}$  &
          UNTR  $\cap$  UNTS= $\emptyset$ , data sample  $X \in \mathbb{R}$  where  $\{x_1, x_2 \dots x_{68}\}$ 
Output: Z // Mean Square Error

Step 1: /*Initialization
          h [ ][] <- 0 // output of each neurons of each hidden layer
           $\hat{x}_i [ ] [ ]$  <- 0 //reconstructed data
          batch_size =256 // number of samples per each iteration
          bias [ ] // bias vector of each hidden layer
          epoch=10 // number of passes to execute learning algorithm for entire dataset
          learning_rate =0.001 // weight adjusting control variable w.r.t gradient
          mse_sum=0, mean_loss=0

Step 2: create and configure the stacked autoencoder network with Adam optimizer
          Step 2.2: for e  $\in$  1 to epoch do
              Step 2.2.1: for b  $\in$  1 to b do
                  n=neurons_structure [1]; // number of neurons for the first layer
                  W = 0 // weight vector W= {w1, w2...wn}
                  define h[1] [ ] = f (WX +bias) // First hidden layer
                  for l in 2 to L do
                      n=neurons_structure[l];
                      W= 0 // weight vector W= {w1, w2...wn}
                      bias =1 // bias value
                      h[l] [ ] = f (W * hl-1+bias) // compute hl from hl-1 of both
                                                                encoder and decoder
                  end for
              end for
          Step 2.2.2  $\hat{x}_i [ ] [ ]$  =h[l] [ ]
                  for I in 1 to  $\hat{x}_i$  do
                      Z <-  $\frac{1}{N} \sum_{i=1}^N ||x_i - \hat{x}_i [ ] [ ]||^2$ 
                      sum = mse_sum + Z
                  end for
          update layer parameters  $\theta^1 = (W, B)$  s. t Z is minimum
          end for

Step 4: mean_loss =  $\frac{mse\_sum}{e}$ 

Step 5: return mean_loss
    
```

It is observed from the above table that the optimized MSE for majority of the activation functions is w.r.t to neurons structure 68-50-30-50-68. Hence this neurons structure is considered to reduce the dimensionality to 30. Once better optimized structure is identified then series of experiments are conducted for dimension reduction on training and testing data.

The inner most hidden layer i.e. last layer of encoder provides the richer representation of CICIDS2017 i.e. 30 features with reduction of 44% of data. Evaluation of the classification efficiency of various activation functions used for dimension reduction are evaluated with SVM-RBF classification model. The next section explores the classification evaluation model.

TABLE IV. MEAN SQUARE ERRORS FOR SOME OF THE STRUCTURES OUT OF THE CONDUCTED EXPERIMENTS

Neurons Structure\ Activation function	Leaky ReLU	Tanh	Linear	ELU	Sigmoid	Softplus
SAE(68-34-17)	0.00007	0.000139	0.000173	0.000072	0.000217	0.000123
SAE(68-50-40-30-10-5)	0.000324	0.0003	0.003887	0.000208	0.002995	0.000851
SAE(68-50-35-20)	0.000373	0.000082	9.52548E-05	0.000039	0.00083	0.000091
SAE(68-50-40-30-20)	0.00028	0.000083	0.000096	0.000039	0.002366	0.000498
SAE(68-34-17-8)	0.000276	0.000284	0.001656	0.000166	0.001296	0.000249
SAE(68-50-40-30)	0.001802	0.000053	2.07121E-05	0.000031	0.00051	0.000208
SAE(68-40-20)	0.000052	0.000091	0.071884	0.000042	0.000224	0.000114
SAE(68-50-30)	0.000042	0.000048	0.00002014	0.000121	0.000136	0.000091

C. SVM-RBF Classification Model

In this module SVM classifier with RBF kernel is used for multi-class classification. Initially SVM-RBF base model is trained and tested with default values of c and gamma. From the existing literature it has been found that the usage of SVM-RBF model in the context of multiclass classification is more suitable than any other existing classifier. Complete dataset which contains 68 features after due preprocessing has been given as an input to the python program with scikit library intended for dimension reduction. Later SVM-RBF classifier is trained with derived reduced datasets of six different SAE models using different activation functions which would serve as a thorough evaluation of the activation functions along with the corresponding confusion matrices. Further the standard performance metrics vis-a-vis accuracy, precision, recall and F-measure are calculated. These experiments were conducted on an experimental setup, Intel core i5 with 1.80 GHz processor with 8GB RAM, windows 10.

V. EXPERIMENTAL RESULTS AND DISCUSSION

This section discusses about the effect of six activation functions pertaining to SAE with the adoption of MSE as a loss function for dimension reduction. Further comparison of different classification metrics which are derived through SVM-RBF classifier is done. The experimental results w.r.t various performance metrics and computational time are given in below Tables V to VII as well as from Fig. 4 to 8.

TABLE V. VARIATION OF ACCURACY IN TRAINING AND TESTING FOR DIFFERENT ACTIVATION FUNCTIONS

Activation Function	Training Accuracy	Testing Accuracy	Average of Training and Testing Accuracy
Without Feature Selection	96.15	96.2	96.175
Leaky ReLU	96.89	96.96	96.925
tanh	96.98	97.05	97.015
Linear	97.19	97.26	97.225
ELU	97.26	97.33	97.295
Sigmoid	97.36	97.38	97.37
Softplus	97.33	97.4	97.365

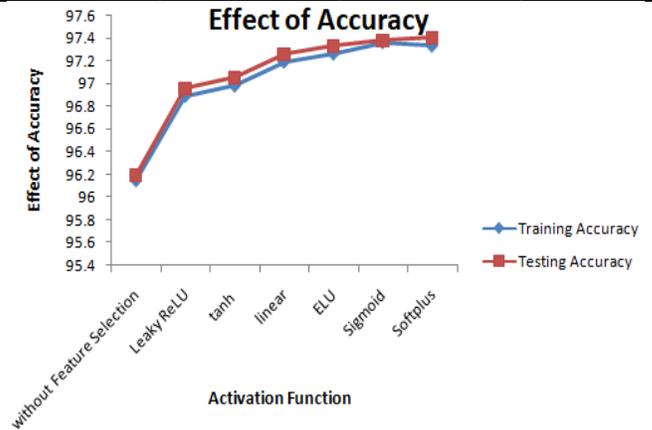


Fig. 4. Effect of the Accuracy of Training and Testing for different Activation Functions.

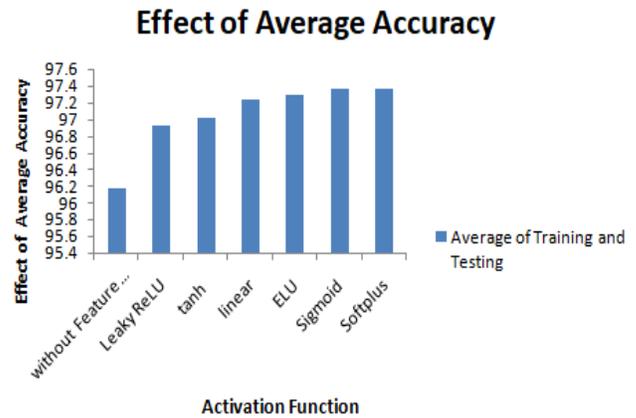


Fig. 5. Effect of Average Computational Time of Training and Testing Time for different Activation Functions.

It is observed that from Fig. 4 and 5:

- The six activation functions exhibits similar behavior for both training and testing.
- Amongst the six softplus gives better accuracy.
- Sigmoid and ELU follows softplus with a minor difference of 0.02 and 0.07, respectively.

TABLE VI. VARIATION OF COMPUTATIONAL TIME IN TRAINING AND TESTING FOR DIFFERENT ACTIVATION FUNCTIONS

Activation Function	Training Time(Sec.)	Testing Time (Sec.)	Average of Training and Testing(Sec.)
Without Feature Selection	24690.515	1123.724	12907.12
Leaky ReLU	4948.109	393.355	2670.732
tanh	7254.742	401.517	3828.1295
Linear	7587.699	302.599	3945.149
ELU	3115.355	306.596	1710.9755
Sigmoid	9128.278	364.093	4746.1855
Softplus	3837.29	338.254	2087.772

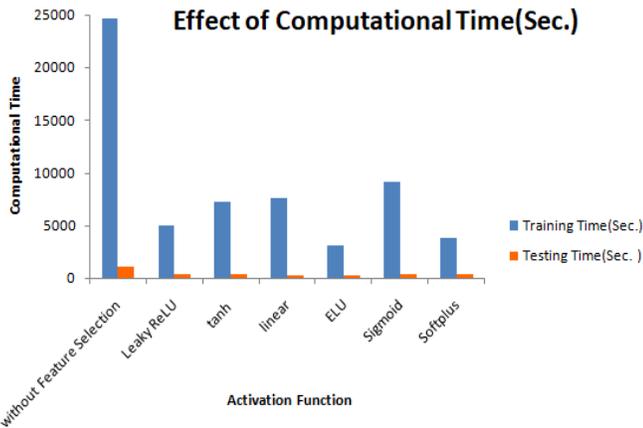


Fig. 6. Effect of the Computational Time of Training and Testing Time for different Activation Functions.

Effect of Average Computational Time(Sec.)

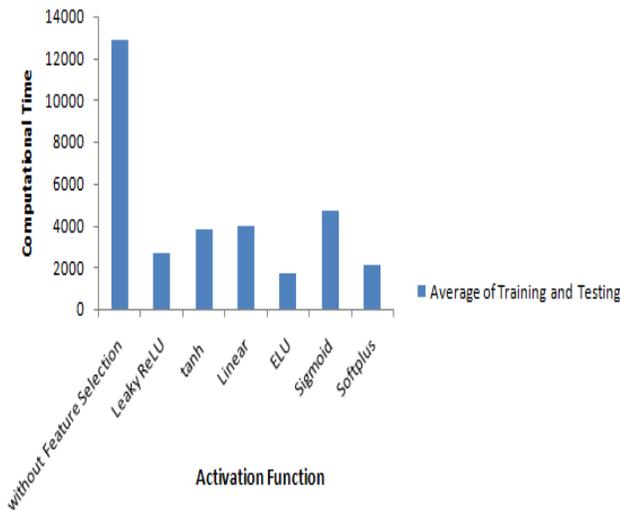


Fig. 7. Effect of Average Computational Time of Training and Testing Time for different Activation Functions.

The computational time of different methods of activation functions are presented in Table VI and Fig. 6 and 7. It is observed that the training and testing time of ELU activation

function is minimum when compared to remaining activation functions. The activation function sigmoid takes highest total execution time. The other two activation functions tanh and linear exhibit more or less same computation time. Therefore ELU activation function must be the choice of researchers to reduce total training time and testing of SVM-RBF classifier using CICIDS2017 dataset.

The experimental results for metrics Precision, Recall, F-measure are presented in Table VII and Fig. 8. In case of precision linear and ELU activation functions exhibits highest performance with a value 0.97 and remaining all four activation function Leaky ReLU, tanh, Sigmoid and Softplus are on the lower side with a value of 0.96. The recall value of linear and ELU activation functions are higher and equal compared to other methods with a value of 0.95. The activation function linear shows next highest value with 0.95 followed by Sigmoid and Softplus which show with an equal value of 0.94. The activation functions linear and ELU obtained best measure results compared to other activation functions. It can be observed that the remaining activation functions are performing more or less equally with a minor difference ranging between 0.1 and 0.5.

TABLE VII. VARIATION OF PRECISION, RECALL AND F-MEASURE ACCURACY IN TRAINING AND TESTING FOR DIFFERENT ACTIVATION FUNCTIONS

Activation Function	Precision	Recall	F-measure
Without Feature Selection	0.79	0.68	0.72
Leaky ReLU	0.96	0.88	0.91
tanh	0.96	0.93	0.95
Linear	0.97	0.95	0.96
ELU	0.97	0.95	0.96
Sigmoid	0.96	0.94	0.95
Softplus	0.96	0.94	0.95

Effect of Precision, Recall and F-Measure

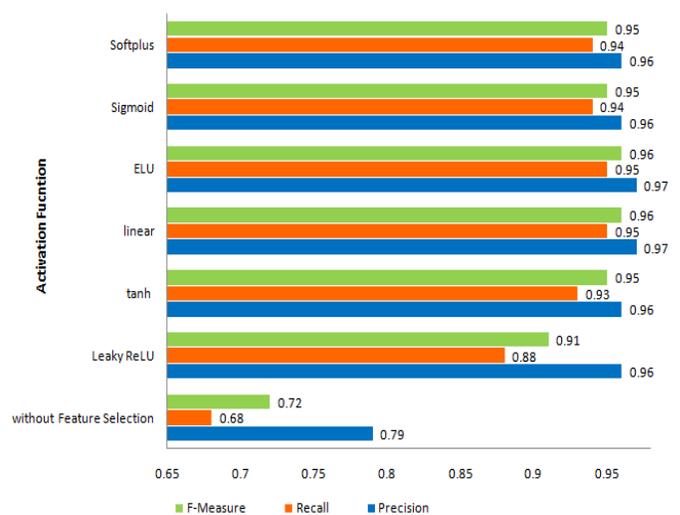


Fig. 8. Effect of Precision, Recall and F-Measure for different Activation Functions.

VI. CONCLUSION

This paper is intended to evaluate the SAE model for dimension reduction through six activation functions. The Experimental results exhibit that the activation function ELU gives better performance in terms of computational time. In the context of precision, recall and F-measure ELU and linear activation functions leads other functions. When classification accuracies are compared the Softplus yields marginal performance. From these experimental results it is observed that ELU is a better activation function with respect to computational time. Whereas ELU and linear provides better performance with respect to other performance metrics. To consider both computational time and classification performance the ELU is better with compromising of negligible difference of accuracy. Finally it is concluded that this comparative study will be of great help to the defenders to design suitable framework for NIDs on cloud environment to defend the intruders within a stipulated time. The extension for this study could be to compare the performance evaluation of different kernel functions of SVM with conducting of more experiments. As an enhancement of this study one can think about evaluating the performance metrics on a real time environment.

REFERENCES

- [1] Ahmed, H. A. S., Ali, M. H., Kadhun, L. M., Zolkipli, M. F., & Alsariera, Y. A. (2017). "A review of challenges and security risks of cloud computing". *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(1-2), 87-91.
- [2] Khalil, I. M., Khreishah, A., & Azeem, M. (2014). "Cloud computing security: a survey". *Computers*, 3(1), 1-35.
- [3] Bulla, S., Rao, B. B., Rao, K. G., & Chandan, K. (2018). "An experimental evaluation of the impact of the EDoS attacks against cloud computing services using AWS". *International Journal of Engineering & Technology*, 7(1.5), 202-208.
- [4] Liu, H., & Lang, B. (2019). "Machine learning and deep learning methods for intrusion detection systems: A survey". *applied sciences*, 9(20), 4396.
- [5] Goswami, S., & Chakrabarti, A. (2014). "Feature selection: A practitioner view". *International Journal of Information Technology and Computer Science (IJITCS)*, 6(11), 66.
- [6] Attallah, O., Sharkas, M. A., & Gadelkarim, H. (2020). "Deep Learning Techniques for Automatic Detection of Embryonic Neurodevelopmental Disorders". *Diagnostics*, 10(1), 27.
- [7] Bhati, B. S., & Rai, C. S. (2020). "Analysis of support vector machine-based intrusion detection techniques". *Arabian Journal for Science and Engineering*, 45(4), 2371-2383.
- [8] Chigozie Enyinna Nwankpa et al, 2018, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning".
- [9] Feng, J., & Lu, S. (2019, June). "Performance analysis of various activation functions in artificial neural networks". In *Journal of Physics: Conference Series* (Vol. 1237, No. 2, p. 022030). IOP Publishing.
- [10] Xu, B., Wang, N., Chen, T., & Li, M. (2015). "Empirical evaluation of rectified activations in convolutional network". *arXiv preprint arXiv:1505.00853*.
- [11] Serwa, A. (2017). "Studying the Effect of Activation Function on Classification Accuracy Using Deep Artificial Neural Networks". *Journal of Remote Sensing & GIS*, 6(03), 6-11.
- [12] Lau, M. M., & Lim, K. H. (2017, April). "Investigation of activation functions in deep belief network". In *2017 2nd international conference on control and robotics engineering (ICCRE)* (pp. 201-206). IEEE.
- [13] Kunang, Y. N., Nurmaini, S., Stiawan, D., Zarkasi, A., & Jasmir, F. (2018, October). "Automatic Features Extraction Using Autoencoder in Intrusion Detection System". In *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)* (pp. 219-224). IEEE.
- [14] Pedamonti, D. (2018). "Comparison of non-linear activation functions for deep neural networks on MNIST classification task". *arXiv preprint arXiv:1804.02763*.
- [15] Alrawashdeh, K., & Purdy, C. (2018, May). "Fast activation function approach for deep learning based online anomaly intrusion detection". In *2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)* (pp. 5-13). IEEE.
- [16] Catak, F. O., & Mustacoglu, A. F. (2019). "Distributed denial of service attack detection using autoencoder and deep neural network". *Journal of Intelligent & Fuzzy Systems*, 37(3), 3969-3979.
- [17] Sstla, V., Kolli, V. K., Voggu, L. K., Bhavanam, R., & Vallabhasoyula, S. "Predictive Model for Network Intrusion Detection System Using Deep Learning Predictive Model for Network Intrusion Detection System Using Deep Learning".
- [18] Le, T. T. H., Kim, J., & Kim, H. (2016). "Analyzing Effective of Activation Functions on Recurrent Neural Networks for Intrusion Detection". *Journal of Multimedia Information System*, 3(3), 91-96.
- [19] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018, January). "Toward generating a new intrusion detection dataset and intrusion traffic characterization". In *ICISSP* (pp. 108-116).
- [20] Abdulhammed, R., Musafar, H., Alessa, A., Faezipour, M., & Abuzneid, A. (2019). "Features dimensionality reduction approaches for machine learning based network intrusion detection". *Electronics*, 8(3), 322.
- [21] El Mrabet, Z., Ezzari, M., Elghazi, H., & El Majd, B. A. (2019, March). "Deep Learning-Based Intrusion Detection System for Advanced Metering Infrastructure". In *Proceedings of the 2nd International Conference on Networking, Information Systems & Security* (pp. 1-7).
- [22] Farahnakian, F., & Heikkonen, J. (2018, February). "A deep auto-encoder based approach for intrusion detection system". In *2018 20th International Conference on Advanced Communication Technology (ICACT)* (pp. 178-183). IEEE.