International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I4P224
Volume 69 Issue 4, 169-176, April 2021

Performance Analysis of Different Activation and Loss Functions of Stacked Autoencoder for Dimension Reduction for NIDS on Cloud Environment

[1]Nirmalajyothi Narisetty, [2]Gangadhara Rao Kancherla, [3]Basaveswararao Bobba, [4]K. Swathi

[1]*Research Scholar, Dept. of CSE, Acharya Nagarjuna University, Guntur, 522510, India.*

[2]*Professor, Dept. of CSE, Acharya Nagarjuna University, Guntur, 522510, India.*

[3]*Dept. of CSE, Acharya Nagarjuna University, Guntur, 522510, India.*

[4]*Associate Professor, Dept. of CSE, NRI Institute of Technology, Agiripalli, 521212, India.*

[1]nirmala.narisetty@gmail.com

Abstract: *This paper's objective is twofold, i) to provide a framework for a better intrusion detection system with an SVM classifier to detect new types of attacks in a cloud environment. ii) Performance comparative study is carried out to identify the best combination of Stacked Autoencoder (SAE) activation and loss functions for dimension reduction. To achieve the first objective, the CICIDS2017 is considered because it consists of modern attacks on Cloud environment-related. The Stacked Autoencoder with backpropagation and Adam Optimizer algorithms meet the second objective of this paper. For this purpose, to conducting experiments, three activation functions and two-loss functions are considered. The Activation functions Rectified Linear Unit (ReLU), SoftMax, and Scaled Exponential Linear Unit (SeLU) are being used as input/hidden and output layers. For loss functions, Mean Squared Error (MSE) and Cross-Entropy (CE) are chosen. To find the effect of these functions' performance metrics, accuracy, precision, recall, f-measure, and computational time are evaluated with SVM classifier using CICIDS 2017 benchmark dataset. The experimental results show that the ReLU-ReLU-CE yields better accuracy, and the SeLU-SeLU-MSE executes with less computational time.*

Keywords: *Auto-encoder, cloud computing, dimensionality reduction, intrusion detection system, machine learning*

I. INTRODUCTION

Cloud Computing (CC) brings exponential growth of accessing cloud services beneficially through Internet and communication/storage technologies at considerably low cost. Still, it also brings about security issues with a new type of attacks[1] like Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. These attacks are continuously generated by intruders who overload the network resources, so that cloud services become unavailable to legitimate users and partners [3]. Since the cloud environment is elastic, additional resources are made available without any human intervention. However, the customer must pay extra money for them, causing a special DDoS attack called EDoS attack.

Consequently, it becomes a challenge in the adoption of cloud computing [3]. The cloud environment's resources and services are affected by such attacks and lead to the violation of the Service Level Agreement (SLA). Based on SLA, cloud resources are provided to the customer, and then resource utilization (e.g., RAM, disk storage) and the computing power are billed to the client [2].

Any approach to prevent this malicious problem may restrict or control the resource allocation, which leads to constrained usage of the Cloud. The complete prevention or eradication of Cloud Computing attacks is not possible; however, some concentrated effort could be used to mitigate them [4]. Therefore, cloud providers need to utilize an intelligent Intrusion Detection System (IDS) to secure and efficient cloud computing operations against such types of attacks.

IDS are categorized based on how they track attacks. They are either founded on signatures or anomalies. By following established identities and patterns, signature-based IDS fits well for existing types of attacks. The IDS can not detect new attacks if the attack signatures are not present in the IDS database, which is a drawback of these sorts of threats. Anomaly-based IDS, on the other side, detect attacks using data-driven machine learning approaches and have an advantage over signature-based IDS in that they can recognize major threats without prior training or understanding [3].

Machine learning (ML) techniques are most promising in network intrusion detection compared to the statistical models from the last two decades. Several Researchers

proposed and implemented various types of classification algorithms for the detection of various types of attacks. Among these algorithms, the Support Vector Machine (SVM) is extensively applied for the IDS problem because of its capability to identify the attacks with a low misclassification rate and minimum classification time.

Several studies show that efficient feature reduction algorithms play a crucial role in evaluating the performance of ML algorithms, Such as reduced the model training time and storage space. Initially, many researchers developed various frameworks based on supervised models utilizing the statistical and Knowledge-based approaches for feature extraction or selection. Unfortunately, former approaches are less accurate when the extracted data dimensionality is extensive [6]. The subset of ML is Unsupervised Deep learning (DL) approaches have a good potential to achieve effective data representation without loss of information for performance improvement of supervised ML algorithms [7]. Also, the integration of unsupervised and supervised models can greatly improve intrusion detection and classification rates [5]. Recently different Autoencoders are used to carry out the feature extraction with different activation and loss functions. All these studies do not cover the effectiveness of feature extraction methodologies with different combinations of the different activation and loss functions. To fulfill this research gap, this work is carried out to achieve this objective.

An appropriate activation function has a better ability to map data in dimensions, additionally increases the capability of a classifier to enhance performance metrics [8]. The loss or cost function estimates how closely the predicted value to the actual values of training data then updates the model's parameters accordingly in the optimization process. The effect of the activation and loss functions on Autoencoder for dimension reduction in intrusion detection is studied.

This study aims to evaluate an unsupervised Stacked Autoencoder for Dimension reduction with an SVM classifier. For this purpose, three different activation functions and two-loss functions are adopted to conduct experiments with the combination of these functions to evaluate performance metrics.

The rest of the paper is organized as follows. Section 2 contains a brief discussion of the relevant literature of this study. A description of the CICIDS2017 dataset is presented in Section 3. The experimental design of the evolutionary process is explained in section 4. The results analysis is provided based on the experiments in section 5. Finally, the conclusions are drawn in section 6.

II. Literature Review

Since cloud servers are hosted on remote locations, services are provided to users through the Internet. Therefore chances of intrusion are more with the possibility of various types of attacks. Thus network intrusion detection has attracted a lot of interest from the research community in securing cloud services. The incorporation of Machine Learning and Deep Learning methods plays an important role in network intrusion detection. Many researchers have employed these techniques on publicly available benchmark datasets for analyzing the performance of intrusion detection methods and metrics. This section discusses some important previous studies in brief.

Fuad Mat Isa et al. [9] investigated IDS based on support vector machine classifier with decision tree and Pearson correlation-based feature selection. They compared the performance of the proposed method based on KDD 99, NSL-KDD, and CICIDS2017 datasets. Exploratory outcomes show that the effective improvement of accuracy on each of the three datasets. Hence authors concluded that the Pearson Correlation method with Tune Model Hyperparameter is the best feature selection method.

Amer A. Abdurrahman et al. [10] adopted information gain ratio-based feature selection with four machine learning algorithms such as C5.0, SVM, Naïve Bayes, and Random Forest algorithms DDoS attack detection. The experiments were implemented on the CICIDS2017 dataset. Among those, the C5.0 algorithm produces the highest performance with 86.45 % accuracy and a surprisingly SVM false positive rate of 75 %.

Moreover, in [11], Razan Abdulhammed et al. investigated various classifiers such as Random Forest (RF), Bayesian Network, Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA) based on two feature dimensionality reduction methods Auto-encoder (AE) and Principle component Analysis (PCA). A dataset with few records of CICIDS2017 is used in this study. They used a Sparse Autoencoder with two hidden layers for latent representation of CICIDS2017 from 81 to 59 using AE and 81 to 10 using PCA. Among the above-specified classifiers, Random Forest demonstrates significant accuracy in both cases.

In this paper, the writers Neha Gupta et al. [12] present two supervised learning models. The effect of overall results utilizing nine activation functions is investigated using Deep Artificial Neural Network (DNN) and Convolutional Neural Network (CNN) models with both the NSL-KDD or UNSW-NB15 datasets. The activation function(s) that performed the best in terms of training accuracy, validation accuracy, and individual perception, as well as total time spent development/certification models. The quality improvement goals for optimal precision while reducing computational time. To tune the DNN and CNN prototypes for productive IDS, the best amplification value is defined.

In [13], Ranjit Panigrahi et al. presented an in-depth analysis of network intrusion detection dataset CICIDS2017. They have identified some shortcomings with the dataset and

suggested some solutions to counter such problems. They have provided solutions with relevant experiments.

Mahmood Yousefi-Azar et al. [14] adopted an Auto encoder-based feature learning approach for reducing the dimensionality of features for intrusion detection. They used a nonlinear activation function with cross-entropy and log loss error for constructing the coding layer. The latent representations of features are input to various classifiers such as Gaussian Naïve Bayes, K-Nearest Neighborhood, Support Vector Machines, and Extreme Gradient Boosting hyperparameter search techniques to improve the accuracy. Among these classifiers, the Gaussian Naïve Bayes classifier outperforms the others.

In [15], Qinxue Meng et al. authors proposed a Relational Autoencoder model to extract high-level features based on both Data itself and their relationship. Reconstruction loss is calculated using Mean Squared Error (MSE) to evaluate the overall performance of it. The same principle is extended to major autoencoder models and is evaluated on MNIST and CIFAR-10 datasets. Experimental results indicate robust features are generated with loss reconstruction error.

In [16], Rung-Ching Chen et al. proposed a data-mining method called Rough set, which they used to reduce features of KDD cup' 99 from 41 to 29. Then chosen features are given as input to SVM to train the model and test, respectively. The proposed strategy exhibits that RST-SVM yields better results.

In [17], Yao Wang et al. combined supervised and unsupervised methods to detect malicious JavaScript on WebPages. Deep learning unsupervised stacked denoising auto-encoder is used to extract the lower representation of data. Then, supervised methods logistic regression and SVM classifiers are successfully implemented for pattern classification to discriminate the malicious from benign JavaScript.

An Effective and Intelligent Intrusion Detection system is proposed in [18] using Deep Auto-encoders. An unsupervised Deep Auto-encoder-based model is trained on NSL-KDD data and further tested on test data. The proposed method's performance in terms of accuracy, precision, and F-measure outperforms the conventional machine learning models.

Wenjuan Wang et al. [7] proposed a cloud intrusion detection method based on Stacked Denoising Autoencoders (SDAE) with an SVM classifier. Simultaneously, SADE is an unsupervised deep learning algorithm for dimensionality reduction, and SVM is a supervised shallow learning algorithm for the detection of malicious attacks with the adoption of the NSL-KDD dataset. But the NSL-KDD dataset does not contain any new type of cloud environment-related attacks. In this study, the authors do not study the influence of activation/loss functions on dimensionality reduction; they studied only default functions. The performance of the classifier mainly depends upon the capability of the dimension reduction algorithms. The dimension reduction algorithm depends on which activation/loss functions are used. So, there is a necessity to study the impact of the different combinations of these functions. This research problem is taken as a current study to fulfill the above tasks, which the above said authors did not consider. This motivates the study of the impact of Stacked Autoencoder's performance with different activation/loss functions for dimensionality reduction through multi-class SVM classifier. The next section describes the CICIDS2017 dataset instead of NSL-KDD because of the reason that the dataset is outdated

III. Description of CICIDS2017 dataset

CICIDS2017 is a new intrusion database created by [19] that contains benign and recent potential real-world attacks that look and behave like real-world data (PCAPs). It also contains the results of a traffic monitoring using CIC Flow Meter, with marked flows based on the time stamp, source and destination IPs, source and destination ports, routers, and attack (CSV files). The definition's extracted features are also available.

The database, according to the source, is spread through eight separate files and contains five days of usual and attacker traffic information from the Canadian Institute of Cybersecurity. Tuesday, Wednesday, and Thursday morning data are best for developing a multi-class detector model, according to [13]. As a result, this study's classification is based on Wednesday's results. There are 692703 instances and 85 features in total, including one mark column with 6 groups: Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slow loris, and Heartbleed. Shows the distribution of recordings by name.

Table 1 CICIDS2017 Dataset category distribution of data

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

In this section, a novel framework is proposed to provide a comparative study of the experiments conducted for different combinations of the activation/loss functions of the Stacked Autoencoder. For this evaluation of the performance metrics with SVM classifier through RBF kernel function. The framework includes Data preprocessing, Dimensionality Reduction, and Classification modules which are explained below.

A. Data preprocessing

The Preprocessing module involves Data cleaning and Normalization. All the features of the CICIDS2017 dataset have numerical values. So, the Preprocessing operations data cleaning and Normalization are carried out.

a) Data cleaning: Some of the database records have null, according to the reference [20]. Since the Machine Learning algorithms do not consume null values, these records are discarded, and they constitute a small proportion of the overall volume of data connected with each attack. 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, Bwd Avg Bytes, Bwd Avg Packets, Bwd These columns have no bearing on any mark classification calculations on a dataset. As a consequence, these columns have been withdrawn. Infinity/NaN values appear in several of the sample values for the two features Flow Bytes/s and Flow Packets/s. For the purposes of implementing data mining algorithms, these values are replaced with zeros. The decreased dataset is extracted by performing above that the cleaning operations, as shown in Table 2.

Table 2. CICIDS2017 Dataset category distribution of data after data cleaning.

Category	Class	Number of Records
Anomaly	DoS GoldenEye	10,293
	DoS Hulk	230,124
	DoS Slowhttptest	5,499
	DoS slowloris	5,796
	Heartbleed	11
Total Anomaly data	---	251,723
Normal	Benign	439,972
Total	---	691,695

b) Normalization: Due to the domination of feature, higher values on the featureless value, the data analysis may result in poor classification results. Normalization is an essential step before applying machine learning algorithms to eliminate such domination and enhance the quality of data. This transformation converts the range of a given feature into a scale that goes from 0 to 1 by using the Min-Max normalization technique to normalize the features of the reduced CICIDS2017 dataset. The feature c values with a range between cmin and Cmax, then the Normalization is defined by [21] with the equation of Cnor= (Ci - cmin)/(Cmax-cmin), Where Cnor is a normalized value of the ith value of feature C. An illustrative example of Normalization for one record is given below.

0.352941, 0.000319,0, 3.67E-06, 4.90E-06, 9.57E-09, 0.000242, 0.002906, 0.001293, 0, 0.000307, 0.003026, 0.001373, 0, 0.005764, 0.40001, 0.000319,0, 0.000319, 0.000319,0,0,0,0, 6.67E-08,0,0,0,0,0,0, 7.23E-06, 3.67E-06, 8.70E-06, 1.31E-05, 0.004144, 0.000242, 0.002632,0,0,0,0,0,0,1,1,0, 0.023256, 0.003446, 0.001293, 0.001373, 7.23E-06,0, 4.90E-06, 3.67E-06, 9.57E-09, 0.003906, 0.01445,0, 0.344262,0,0,0,0,0,0,0,0,0

Figure 1 Resulting in one sample record after Normalization

B. Dimensionality reduction

Dimension reduction of the feature set is an important preprocessing technique to improve Machine Learning algorithms' efficiency and reduce N. Nirmalajyothi et al.
N. Nirmalajyothi et al.
Computational complexity. Several authors proposed methodologies from the last two decades using statistical, rough, and fuzzy set algorithms for dimension
reduction [22-24]. Autoencoders are also proposed for this purpose when the dataset is nonlinear. Most of these studies are used for building Autoencoder models with fixed activation and loss functions. Still, there is a research gap to identify which combination of the activation and loss functions are better for dimension reduction. To fulfill this research, a gap in this study is carried out to build a stacked Autoencoder with a different combination of nonlinear activation and loss functions. Further, these combinations are compared with the evaluation of the SVM Classifier. The activation functions ReLU, SeLU, and SoftMax, are chosen for input/hidden and output layers, Whereas Mean Squared Error (MSE) and Cross-Entropy are chosen as loss functions. Five experiments have been conducted for dimension reduction intended to identify these functions' effect through performance evaluation of SVM Classifier. The following Table 3 exhibits the various Activation and Loss functions considered for this work.

Table 3. Details of the chosen functions of the Activation/Loss Functions for conducting experiments.

Experiment No.	Name of the Activation unction		Name of the Loss Function	Name of the Experiment for Dimension Reduction
	Input/ Hidden Layer	Output Layer		
1.	ReLU	SoftMax	MSE	ReLU-SoftMax-MSE
2.	SeLU	SoftMax	MSE	SeLU-SoftMax-MSE
3.	ReLU	ReLU	Cross Entropy	ReLU-ReLU-CE
4.	ReLU	ReLU	MSE	ReLU-ReLU-MSE
5.	SeLU	SeLU	MSE	SeLU-SeLU-MSE

C. Classification

The classification module SVM classifier with RBF kernel function is used for multi-class classification of five types of attacks and the benign. The RBF kernel function has two hyperparameters C and σ, with default values of C=1 and σ=0.34. 70 % of the CICIDS2017 benchmark dataset is used for the training model and the rest of the testing dataset. The experiments are repeated for each combination of active and loss functions, as mentioned in Table 3. The flow of the framework is presented in Figure 2

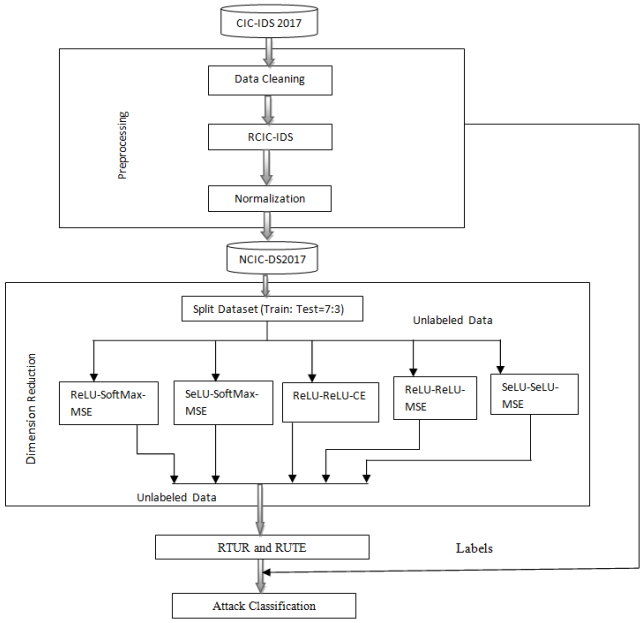

Figure 2. The flow of the proposed framework

D. Experimental Design

This section's main focus is to seek the optimal feature subset with the implementation of Stacked Autoencoder with five layers and evaluate the performance measures with SVM. The unit numbers of these layers are 68, 50, 30, 30, 50, and 68. As per the [25] studies, it is known that Adam Optimizer is better than the stochastic optimization method. So in this study, Adam Optimizer is adopted with backpropagation for iteratively updating network weights. The parameter values are set as Hidden Layers=2, learning rate= 0.01, epochs=10, batch_size=256.

For methodology implementing and conducting of experiments with the environment of Win-10 machine. The configuration is Intel(R) Core (TM) i5- 8250U CPU @1.80 GHz, 8 GB Ram. In this experimental study, dimensionality, reduction, and classification approaches are implemented using an open-source Keras framework containing different libraries for machine learning algorithms. Algorithm 1 shows the pseudo-code of the dimension reduction process using Stacked Autoencoder. The steps from 1 to 5 configure the Stacked Autoencoder for the given activation function. In step6 model is built and trained with training data in step7. If the loss function applied is MSE, then step8 computes loss. Else step9 is invoked. The encoder model is constructed in step10, followed by the same model is employed to reduce both training and test data. Lastly, Algorithm returns the reduced data.

Algorithm1 shows the pseudo-code of the dimension reduction process using Stacked Autoencoder

Algorithm: Dimensionality Reduction on NCICIDS2017 dataset

Input: $\{x_1, x_2 \ldots . x_n\}$ is the normalized n feature set of Unlabeled Training data UTR
and Unlabeled Test data UTE, where n=68

Output: Reduced Unlabeled Training and Testing dataset RUTR and RUTE

Step1. inputlayer<-(units= n)
Step2. E_Layer_1<- Dense (units=50, activation=relu) (input layer)
Step3. E_Layer_2<-Dense (units=30, activation=relu) (E_Layer_1)
Step4. D_Layers_1<-Dense (units=50, activation=relu) (E_Layer_2)
Step5. D_Layers_2<-Dense (units=n, activation=relu) (D_Layers_1)
Step6. FSAE_Model<-Model (input layer, D_Layers_2)
Step7. FSAE_Model.fit (UTR)
Step8. If (loss=MSE)
for i=0 to $\hat{X}$ do
loss <- $\frac{1}{N}\sum_{i=1}^{n} ||x_i - x_i^1||^2$

end for
Step9. else
for i=0 to $\hat{X}$ do
loss <- $(-\sum_{i=1}^{n} X_i\,(logY_i) + (1 - X_i)log(1 - Y_i)$
end for
Step10. encoded<- FSAE_Model.E_Layer$_2$
Step11. RUTR<- []
Step12. RUTR<-encoded (UTR)
Step13. RUTE<- []
Step14. RUTE<- encoded (UTE)
Step15. return RUTR, RUTE

Autoencoder is used in this experiment as an unsupervised model, which utilizes unlabeled data for dimensionality reduction. Afterward, the new representation of the CICIDS2017 dataset from the innermost hidden encoder layer is used to reduce unlabeled data. While the supervised SVM classifier needs labeled data, subsequently, the new representation is combined with labeled data (Yi). SVM classifier with RBF kernel function is used for six-class classification of cloud network traffic due to its effectiveness and good classification accuracy. RBF kernel has two parameters C and σ; the proposed model is tested with their respective default values. 70 % of the CICIDS2017 benchmark dataset is used for the training model and the rest for testing. The classification model is built on training data.

V. RESULTS AND DISCUSSION

This section analyzes the attack detection results obtained by conducting various experiments mentioned in the above section. The performance metrics, namely accuracy, precision, recall, F-measure, and SVM classifier with RBF kernel function, are considered. The loss in Dimensionality reduction for the combinations SeLU-SoftMax-MSE, ReLU-SoftMax-MSE, ReLU-SeLU-CE and SeLU-SeLU-MSE are very high. so these combinations are not considered for the classification.

Table 4. Effect of various performance metrics for a different combination of the activation and loss functions.

Method	Training Accuracy	Testing Accuracy	Precision	Recall	F-M Easure
SVM	96.15	96.2	0.79	0.68	0.72
ReLU-SoftMax-MSE	96.51	96.57	0.79	0.71	0.75
SeLU-SoftMax-MSE	96.51	96.57	0.79	0.74	0.76
SeLU-SeLU-MSE	97.07	97.14	**0.97**	0.94	0.95
ReLU-ReLU-MSE	97.19	97.27	**0.97**	0.94	**0.96**
ReLU-ReLU-CE	**97.3**	**97.38**	**0.97**	**0.95**	**0.96**

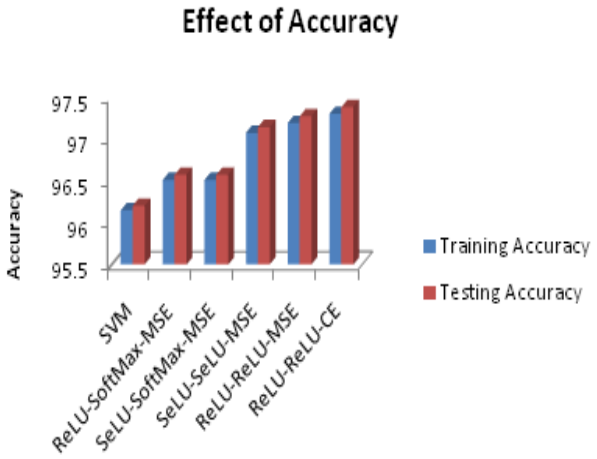

Figure 3 Effect of the accuracy for different combinations of various activation and loss functions.

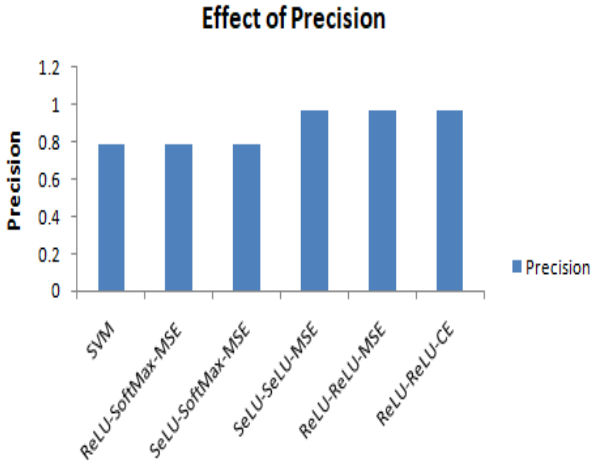

Figure 4 Effect of the precision for different combinations of various activation and loss functions.

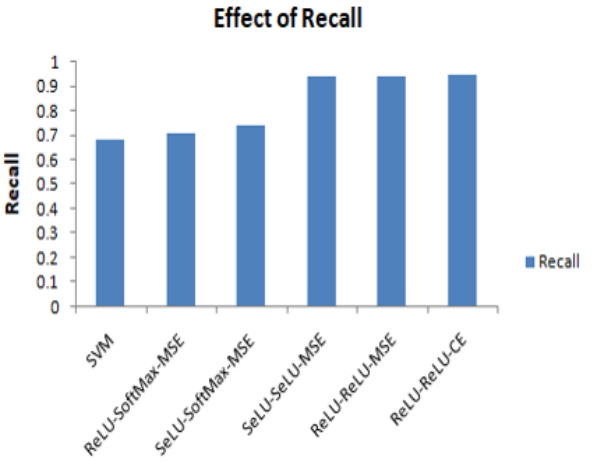

Figure 5 Effect of the recall for different combinations of various activation and loss functions

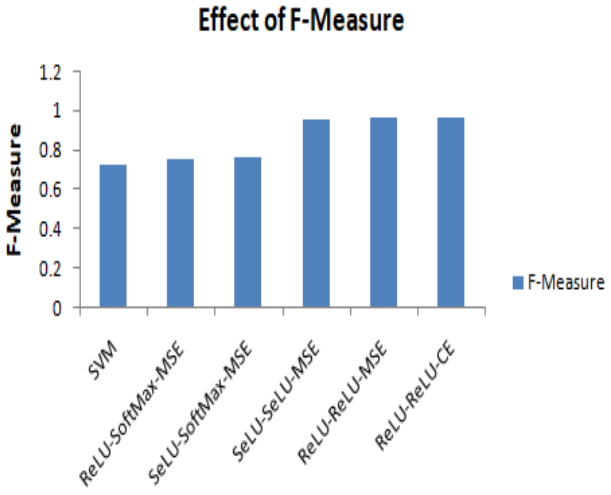

Figure 6 Effect of the F-measure for different combinations of various activation and loss functions.

The following observations are made as per the various performance metrics depicted in Table 4 and Figures from 3 to 6. Both testing and training accuracy exhibited better accuracy with ReLU-ReLU-CE, i.e., 97.38 %, compared to other methods with and without stacked Autoencoder. The accuracies of all methods invariably lie within 1 % difference. In case of Precision SeLU-SeLU-MSE, ReLU-ReLU-MSE and ReLU-ReLU-CE methods have identical values i.e. 0.96. The remaining methods perform with values of 0.79, and there exists a significant difference, i.e., 0.17. The recall value of the ReLU-ReLU-CE method is higher than the remaining methods, with a value of 0.95. The methods SeLU-SeLU-MSE and ReLU-ReLU-MSE are almost all close to ReLU-ReLU-CE with a difference of 0.1. Compared to ReLU-ReLU-CE, the other three methods exhibit less performance with a different value of 0.21, 0.24, and 0.27.

F-measure is the best metric to evaluate the classification model when class distributions of the dataset are imbalanced. The higher the F-measure, the better is the model, which indicates good results. Comparatively, the ReLU-ReLU-MSE, ReLU-ReLU-CE methods exhibit better performance with a value of 0.96. Amongst all the remaining methods, the SeLU-SeLU-MSE is very close to the higher performance methods with a different value of 0.1. The remaining three methods show less performance with a different value of 0.20, 0.21, and 0.24.

Table 5 Snapshot of the Computational time of SVM classifier in comparison with different combinations of activation and loss functions

Method	Training Time(sec.)	Testing Time(sec.)
SVM	24,690.515	1,123.724
ReLU-SoftMax-MSE	5,196.669	479.997
SeLU-SoftMax-MSE	5,169.58	504.303
SeLU-SeLU-MSE	4,310.975	333.333
ReLU-ReLU-MSE	5,112.665	349.604
ReLU-ReLU-CE	9,499.291	355.495

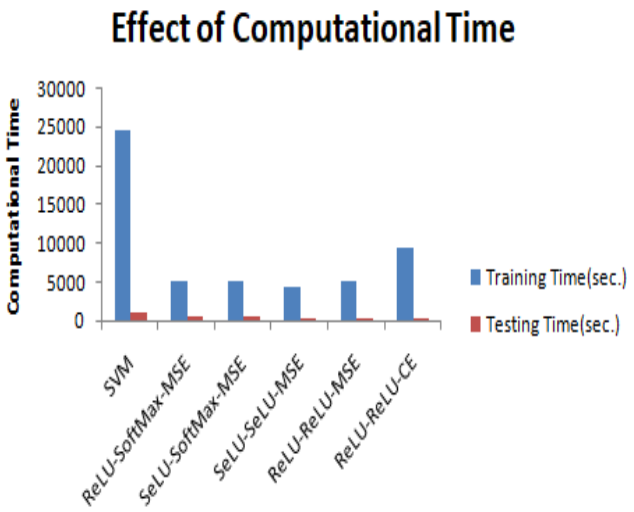

Figure 7 Effect of the Computational time for different combinations of various activation and loss functions

Any Machine Learning algorithm is an effective algorithm compared to other algorithms, if and only if the Algorithm satisfies two conditions. They are i) The performance metrics accuracy, precision, recall, and F-Measure have higher values ii) its computational time is minimum. The computational time of different activation and loss functions is presented in Table 4 and Figure 7. It is observed that SeLU-SeLU-MSE computational time for both training and testing is minimum when compared to other methods. All other remaining methods have been executed with a little bit of difference in the computational time of SeLU-SeLU-MSE, except the ReLU-ReLU-CE method, which takes the highest execution time.

VI. CONCLUSIONS

In this paper, a novel framework is provided to build a classifier for detecting malicious tasks in a cloud environment. For this purpose, an unsupervised Stacked Autoencoder for feature extraction and supervised classifier SVM is adopted. This classifier's performance has been evaluated using the different combinations of the three activation and two-loss functions using CICIDS2017. The Experimental results exhibit both methods; ReLU-ReLU-CE/MSE are almost all given equal performance metric values with a marginal difference. The classification performances of these methods in terms of accuracy, precision, recall, and F-measure are highest compared to other methods. The method SeLU-SeLU-MSE performs with a minimum computational time for training for testing than compared to other methods. Finally, it is concluded that this study will help the defenders for effective designing of Intrusion Detection System on the cloud environment. As a future study to compare the performance evaluation of different kernel functions of SVM with conducting more experiments. As an enhancement of this study to evaluate the performance metrics in a real-time environment.

References

[1] Deshmukh, R. V., & Devadkar, K. K., Understanding DDoS attack & its effect. In cloud environment., Procedia Computer Science, 49 202-210.

[2] Zekri, M., El Kafhali, S., Aboutabit, N., & Saadi, Y., DDoS attack detection using machine learning techniques in cloud computing environments, (2017).

[3] In 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech) IEEE 1-7, (2017).

[4] Kumar, R., Lal, S. P., & Sharma, A., Detecting denial of service attacks in the cloud. In IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech) (2016) 309-316. IEEE.

[5] Abbasi, H., Ezzati-Jivan, N., Bellaiche, M., Talhi, C., & Dagenais, M. R., Machine learning-based EDoS attack detection technique using execution trace analysis, Journal of Hardware and Systems Security, 3(2)(2019) 164-176.

[6] Al-Qatf, M., Lasheng, Y., Al-Habib, M., & Al-Sabahi, K., Deep learning approach combining sparse autoencoder with SVM for network intrusion detection, IEEE Access, 6 (2019) 52843-52856.

[7] Kunang, Y. N., Nurmaini, S., Stiawan, D., Zarkasi, A., & Jasmir, F., Automatic Features Extraction Using Autoencoder in Intrusion Detection System. In International Conference on Electrical Engineering and Computer Science (ICECOS) (2018) 219-224. IEEE.

[8] Wang, W., Du, X., Shan, D., & Wang, N., A Hybrid Cloud Intrusion Detection Method Based on SDAE and SVM, In12th International Conference on Intelligent Computation Technology and Automation (ICICTA) (2019) 271-274. IEEE. DOI 10.1109/ICICTA49267.2019.00064

[9] Wang, Y., Li, Y., Song, Y., & Rong, X., The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition, Applied Sciences, 10(5)(2020) 1897.

[10] Isa, F. M., Buja, A. G., Darus, M. Y., & Saad, S., Optimizing the Effectiveness of Intrusion Detection System by using Pearson Correlation and Tune Model Hyper Parameter on Microsoft Azure Platform. International Journal, 9(1.3)(2020).

[11] Abdulrahman, A. A., & Ibrahem, M. K., Evaluation of DDoS attacks Detection in a New Intrusion Dataset Based on Classification Algorithms., Iraqi Journal of Information & Communications Technology, 1(3)(2018) 49-55.

[12] Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., & Abuzneid, A., Features dimensionality reduction approaches for machine learning-based network intrusion detection., Electronics, 8(3) (2019) 322.

[13] Gupta, N., Bedi, P., & Jindal, V., Effect of Activation Functions on the Performance of Deep Learning Algorithms for Network Intrusion Detection Systems, In Proceedings of ICETIT (2019) 949-960. Springer, Cham.

[14] Panigrahi, R., & Borah, S.., A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems., International Journal of Engineering & Technology, 7(3.24)(2018) 479-482.

[15] Yousefi-Azar, M., Varadharajan, V., Hamey, L., & Tupakula, U. (2017)., Autoencoder-based feature learning for cybersecurity applications., In 2017 International joint conference on neural networks (IJCNN) 3854-3861. IEEE.

[16] Meng, Q., Catchpoole, D., Skillicom, D., & Kennedy, P. J. 2017, May)., Relational autoencoder for feature extraction., In 2017 International Joint Conference on Neural Networks (IJCNN) 364-371. IEEE.

[17] Chen, R. C., Cheng, K. F., Chen, Y. H., & Hsieh, C. F., Using rough set and support vector machine for network intrusion detection system. In 2009 First Asian Conference on Intelligent Information and Database Systems (465-470) IEEE.

[18] Wang, Y., Cai, W. D., & Wei, P. C., A deep learning approach for detecting malicious JavaScript code. Security and Communication Networks, 9(11)(2016) 1520-1534.

[19] POONGOTHAI, T., & JAYARAJAN, K., An Effective and Intelligent Intrusion Detection System using Deep Auto-Encoders.

[20] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A., Toward generating a new intrusion detection dataset and intrusion traffic characterization., In ICISSP (2018) 108-116.

[21] ABDULRAHEEM, M. H., & IBRAHEEM, N. B., A DETAILED ANALYSIS OF NEW INTRUSION DETECTION DATASET., Journal of Theoretical and Applied Information Technology, 97(17)(2019).

[22] Krishna, K. V., Swathi, K., & Rao, B. B., A Novel Framework for NIDS through Fast kNN Classifier on CICIDS2017 Dataset.

[23] Banerjee, M., Mitra, S., & Anand, A., Feature selection using rough sets., In Multi-Objective Machine Learning (2006) 3-20. Springer, Berlin, Heidelberg.

[24] Caballero, Y., Alvarez, D., Bello, R., & Garcia, M. M., Feature selection algorithms using rough set theory., In Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007) 407-411. IEEE.

[25] Shen, Z., Chen, X., & Garibaldi, J., Performance optimization of a fuzzy entropy-based feature selection and classification framework., In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)(2018) 1361-1367. IEEE.

[26] Kingma, D. P., & Ba, J., Adam: A method for stochastic optimization., arXiv preprint arXiv:1412.6980., (2014).

[27] Ahmad Akl, Ahmed Moustafa, Ibrahim El-Henawy, Deep Learning: Approaches and Challenges, International Journal of Engineering Trends and Technology 65(1) (2018) 9-16.