Journal of Multimedia and Information System VOL. 3, NO. 3, September 2016(pp. 91-96): ISSN 2383-7632(Online)
http://dx.doi.org/10.9717/JMIS.2016.3.3.91

Analyzing Effective of Activation Functions on Recurrent Neural Networks for Intrusion Detection

Thi-Thu-Huong Le[1], Jihyun Kim[2], Howon Kim[3,*]

Abstract

Network security is an interesting area in Information Technology. It has an important role for the manager monitor and control operating of the network. There are many techniques to help us prevent anomaly or malicious activities such as firewall configuration etc. Intrusion Detection System (IDS) is one of effective method help us reduce the cost to build. The more attacks occur, the more necessary intrusion detection needs. IDS is a software or hardware systems, even though is a combination of them. Its major role is detecting malicious activity. In recently, there are many researchers proposed techniques or algorithms to build a tool in this field. In this paper, we improve the performance of IDS. We explore and analyze the impact of activation functions applying to recurrent neural network model. We use to KDD cup dataset for our experiment. By our experimental results, we verify that our new tool of IDS is really significant in this field.

Key Words: Recurrent Neural Networks, activation function, non-linear activation, deep learning, Intrusion Detection System.

I. INTRODUCTION

The Internet which varies from the source with useful information has become more and more developing. Hence, the data across the Internet must be secure. Intrusion Detection System (IDS) is, therefore, the tool needed in for this requirement. IDS became an essential part of the security management since network administrator based on IDS in order to prevent malicious attacks. Furthermore, IDS can detect and block attacks on the network, retain the performance normal during any malicious outbreak, perform an experienced security analysis. IDS have two different classified groups. First is anomaly detection and other is signature based detection or misused detection. To detect unknown anomalies we used to anomaly detection. On the other hand, misused detection used the known pattern to detect attacks. However, misused detection cannot detect the unknown anomalies. There are many researchers have done significant this work to develop IDS. These papers proposed an architecture of the classification techniques, algorithms being used such as Machine Learning techniques and the algorithms of neural networks.

In this paper, we organized as follows: Section II introduce the related work. Section III presents briefly background recurrent neural network and activation functions. Experiential setup is shown in Section IV. Section V is the result of the experiment. Final Section includes conclusion for our work.

II. RELATED WORK

There are many approaches applied for IDS based on Machine Learning. An experimental framework to compare supervised (classification) and unsupervised (clustering) learnings for detecting attack activities by Laskov [1]. The results of [1] show that the supervised algorithms show better classification accuracy on the data with known attacks. Besides Lee at el. [2] build a classifier to detect anomalies in networks using data mining techniques.

In addition, there are several algorithms based on four techniques of computational intelligent: genetic algorithms (GA), artificial neural networks (ANN), fuzzy logics (FL) and artificial immune system (AIS). To GA, Sinclair [3] use genetic algorithms and decision tree to create rules for intrusion detection expert system. Then, Li [4] describes a few disadvantage of the algorithm in [3] and defines new

Manuscript received July 2, 2016; Revised July 15; Accepted July 28,2016. (ID No. JMIS-2016-0008)
Corresponding Author (*): Howon Kim, Pusan National University, PusanDaeHakRo 63BeonGil-2 (JangJeon-Dong), GeumJeong-Gu, Busan, Republic of Korea, +82-51-510-1010, howonkim@pusan.ac.kr.
[1]School of Computer Science and Engineering, Pusan National University, Busan, Korea, lehuong7885@gmail.com
[2] School of Computer Science and Engineering, Pusan National University, Busan, Korea, jihyunkim@pusan.ac.kr
[3] School of Computer Science and Engineering, Pusan National University, Busan, Korea, howonkim@pusan.ac.kr

techniques for IDS rules. Besides ANN have the ability to learning by example and generalize from limited, noisy, and incomplete data. By combining neural network and SVM Mukkamala [5] applied it to intrusion detection. To FL, Gomer and Dasgupta [6] show that with fuzzy logic, the false alarm rate in determining intrusive activities can be reduced. Finally, AIS which consist of molecules, cells, and tissues that establish body's resistance to infection by pathogens like bacteria, viruses, and parasites. The first of AIS is modelled various computer security problems by Hofmeyr [7]. Then, Kim [8] provide key developments computer security and six immune features for an effective IDS. Zamani [9] [10] describe an artificial immune algorithm for IDS by proposed a multi-agent environment the computationally emulates the behavior of the natural immune system to reduce false positive rates.

III. BACKGROUND

In this section, we summary about the knowledge of Recurrent Neural Networks (RNN) and activation functions.

3.1 Recurrent Neural Network

RNN is an extension from Feed Forward Neural Network. RNNs are called recurrent since they perform the same task for every element of a sequence, with the output being depended on the previous computations.

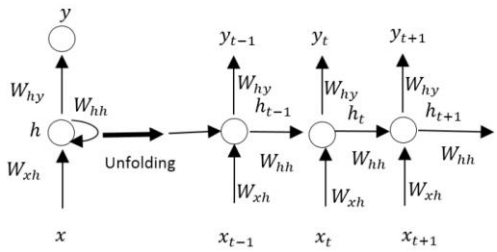


Fig 1. A recurrent neural network and the unfolding in time of its forward computation.

Fig.1 is a typical RNN and the unfolding in time of the computation involved in its forward computation. An RNN consists of three layers. The first is input layer (x). The second is hidden layer (h). Output layer (y) is the final.

First, we need to calculate hidden layer at time t based on the previously hidden state and the input at the current step:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1}) \quad (1)$$

Second, we need to compute the prediction of the model from hidden layer to output layer.

$$y_t = f(W_{hy}h_t) \quad (2)$$

Where:

- W_{xh} is weight matrix connecting between input layer to hidden layer
- W_{hh} is weight matrix connecting between hidden layer to hidden layer
- W_{hy} is weight matrix connecting between hidden layer to output layer
- f is an activation function, we describe in Section 3.2.

3.2 Activation functions

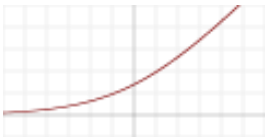
Activation functions transform neuron's input into output layer. Features of activation functions consist of two issues. Firstly, it is a squashing effect is required. It prevents accelerating the growth of activation levels through the network. Secondly is simple and easy to calculate. There are many activation functions that are used to for neural network. However, we choose several representative activation functions which effect to train RNN model. They are nonlinear activation functions. We briefly describe these activation functions follow as:

- SoftPlus

This activation function was proposed by Glorot [11]. This softplus function can be approximated by max function (or hard max), i.e. $\max(0, x + N(0,1))$. The gradient of the sigmoid function vanishes as we increase or decrease x. The equation of SoftPlus is:

$$f(x) = \log_e(1 + e^x) \quad (3)$$

Here is the plot of this activation

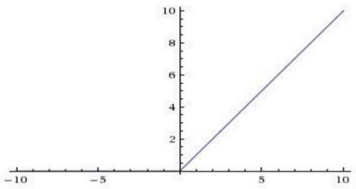


- ReLU (Rectified Linear Unit)

ReLU was proposed by Nair [12]. ReLU does not face gradient vanishing problem as with sigmoid and tanh function. Also, it has been shown that deep networks can be trained efficiently using ReLU even without pre-training. The equation of this function is:

$$f(x) = \begin{cases} 0 \ for & x < 0 \\ x \ for & x \geq 0 \end{cases} \quad (4)$$

Here is the plot of ReLU activation:

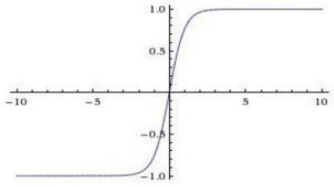


- Tanh

Tanh is a transformed version of Sigmoid which takes values in ± 1 instead of the unit interval. Input with large absolute values and approximate 1 for large positive inputs. The equation of this function is:

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \quad (5)$$

Here is the plot of Tanh activation:

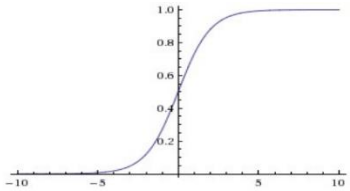


- Sigmoid

Sigmoid function has range [0, 1]. Hence this function can be used to model probability. The equation of this function is:

$$f(x) = sigmoide(x) = \frac{1}{1+e^{-x}} \quad (6)$$

Here is the plot of this activation:

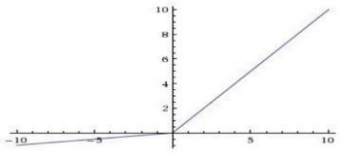


- LeakyReLU (Leaky Rectified Linear Unit)

This activation is first introduced in the acoustic model by Maas [13]. A LeakyReLU can help fix the dying ReLU problem. ReLU's can die if a large enough gradient changes the weights such that the neuron never activates on new data. We have the equation:

$$f(x) = \begin{cases} x_i & if \quad x_i \geq 0 \\ \frac{x_i}{a_i} & if \quad x_i < 0 \end{cases} \quad (7)$$

Here is the plot of LeakyReLU:

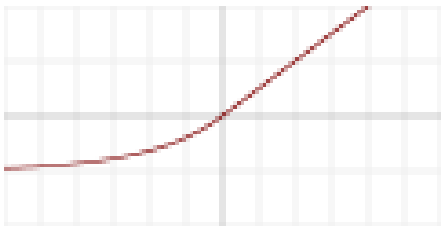


- ELU (Exponential Linear Unit)

This activation function is proposed by Clevert [14]. This function which speeds up learning in deep neural networks. ELU alleviate the vanishing gradient problem via the identity for positive values. The formula for this function is:

$$f(x) = \begin{cases} \alpha(e^x - 1) \, if & x_i < 0 \\ x \quad if \; x_i \geq 0 \end{cases} \quad (8)$$

The plot of ELU is shown in graph below:



V. EXPERIMENTAL SETUP

There are several main points which we need to mention in our processing. The first, we need to choose a suitable dataset to experiment. The second, we set up the best of values for model's hyperparameters. The third, the implementing environment is set up for training and testing processes. The final, we use to evaluation metrics to evaluate our training model.

4.1 Dataset description

We choose KDD Cup dataset to train our model. There are four types of category attack as such as DoS, R2L, U2R, and Probe. Each attack consists of many small attacks. Dos (Denial of Service) is denied legitimate requests to a system. U2R (User-to-Root) is unauthorized access to local super user (root) privileges. R2L (Remote-to-Local) is unauthorized access from a remote machine. Probing (Probe) is surveillance and another probing. Each attack consists of many small attacks. The training dataset has 22 types of attacks is shown in Table 1. And Table 2 shows 37 types of attacks for the testing dataset. How to do preprocessing this dataset for our experiment is mentioned in [15].

Table 1. List of attacks for training dataset.

Name of category	Name of attacks
DoS	back, land, neptune, pod, smurf, teardrop
Proble	ipsweep, nmap, portsweep, satan
R2L	ftpwrite, guespasswd, imap, multihop, phf, spy, warezclient, Warezmaster
U2R	bufferiverflow, loadmodule, perl, rootkit

Table 2. List of attacks for testing dataset.

Name of category	Name of attacks
DoS	Apache2, back, land, mailbomb, Neptune, pod, processtable, smurf, teardrop
Proble	ipsweep, mscan, nmap, portsweep, saint, satan
R2L	ftpwrite, guespasswd, httptunnel, imap, multihop, named, phf, sendmail, snmpgetattack, warezmaster, xlock, xsnoop
U2R	bufferiverflow, loadmodule, perl, ps, rootkit, snmpguess, sqlattack, worm, xterm

4.2 Initiation hyperparameters of model

The hyperparameters of values are important when we train our model. Therefore, choosing these values as suitable will help us achieve better performance. In our work, we set up the hyperparameters to train for our model as the manual. The values are shown in detail in Table 3:

Table 3. The value of hyperparameters.

Name of hyperparameter	Value
Learning rate	0.001
Number of hidden layers	80
Epochs	500

4.3 Implementation environment setup

We perform to measure classification performance on our environment as following: Intel ® CoreTM i7-4790 CPU @3.60GHz; GPU: NIVIA GeForce GTX 750; RAM: 8GB and OS: Windows 7.

4.4 Evaluation metrics

We use to confusion matrix to evaluate our model. There are some metrics to compute such as Accuracy, Precision, and Recall. Here are some equations:

$$recall = \frac{TP}{TP + FN} \quad (9)$$

$$precision = \frac{TP}{TP + FP} \quad (10)$$

$$accuracy = \frac{TP + TN}{(TP + FP) + (FN + TN)} \quad (11)$$

Where,

- TP is a number of predicted as Normal while they actually were Normal.
- FP is a number of predicted as Attack while they actually were Normal.
- FN is a number of predicted as Normal while they actually were Attack.
- TN is a number of predicted as Attack while they actually were Attack.

Besides, we calculate the False Alarm Rate (FAR) which is the ratio of misclassified normal attack.

$$FAR = \frac{FP}{FP + TN} \quad (12)$$

V. EXPERIMENTAL RESULTS

We perform our approach with six activation functions to finding the best activation function for RNN on IDS.

In Fig. 2, we observe the graph that describes the classifying of each attack on IDS with our model using activation functions.

In particular, the result in detail of that graph is shown in Table 4. Among them, the best of the result is at *LeakyReLu* activation with almost the biggest of values.

Besides, we perform performance of classifying on IDS with RNN using six activation functions. By computing the Accuracy, Recall, and Precision, we present our result in Fig.3. And this result is described in Table 5 in detail.

Furthermore, we compute False Alarm Rate (FAR) measurement. The smaller value of FAR, the better our approach. Obviously, the smallest of FAR is near 0.084 at *LeakyReLu* activation.

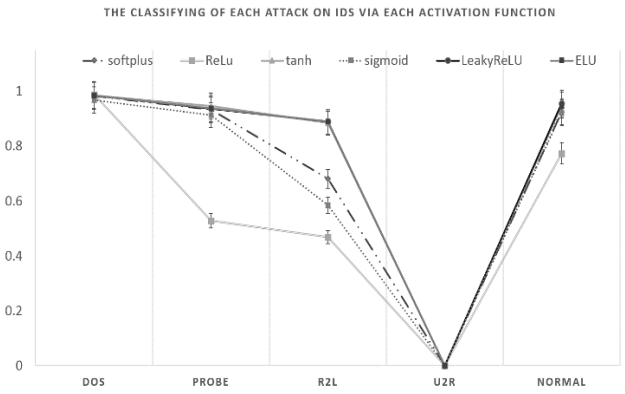


Fig. 2. The classifying of each attack using RNN with activation functions.

Table 4. The classifying of each attack on IDS

RNN with activation function	Attach's name				
	DoS	***Probe***	***R2L***	***U2R***	***Normal***

RNN with activation function	**Attach's name**				
	DoS	***Probe***	***R2L***	***U2R***	***Normal***
softplus	0.983	0.933	0.6808	0	0.9237
ReLu	0.9866	0.5278	0.4677	0	0.7733
tanh	0.9839	0.9453	0.8834	0	0.9492
sigmoid	0.968	0.913	0.5847	0	0.9208
LeakyReLu	**0.9842**	**0.9364**	**0.8885**	**0**	**0.9552**
ELU	0.9837	0.9367	0.8885	0	0.943

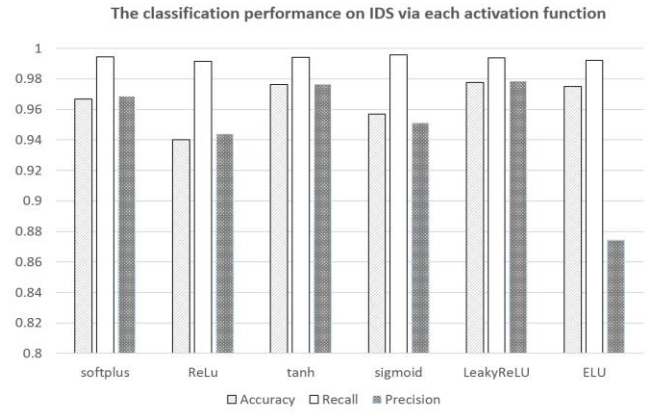


Fig. 3. The classification performance on IDS using RNN with activation functions.

Table 5. The measuring metrics of performance

RNN with activation function	**Measuring metrics of performance**		
	Accuracy	***Recall***	***Precision***
softplus	0.9669	0.9943	0.9686
ReLu	0.9401	0.9915	0.9439
tanh	0.9764	0.9941	0.9765
sigmoid	0.957	0.9958	0.9513
LeakyReLu	**0.9777**	**0.9938**	**0.9785**
ELU	0.975	0.9921	0.8742

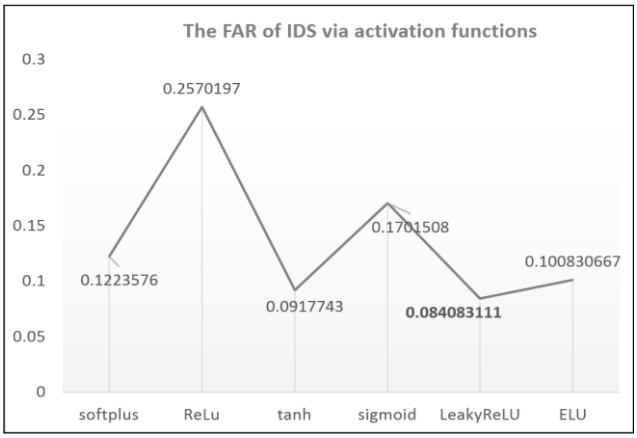


Fig. 4. The FAR of IDS using RNN with activation functions.

From these results, we conclude *LeackyReLu* activation is used to RNN outperform to others.

VI. CONCLUSION

In our work, we perform our approach using RNN model though six activation functions. By our results, we found that LeakyReLU function returns to the best of performance classification among them. Particularly, we achieve 97.77%, 87.85% and 99.38% to accuracy, precision and recall. Hence, we confirm that RNN model using LeakyReLU function can build a new better IDS classifier.

Acknowledgement

This paper is supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 10043907, Development of high performance IoT device and Open Platform with Intelligent Software).

REFERENCES

[1] P. Laskov et al., "Learning intrusion detection: Supervised or unsupervised?," *In Image Analysis and Processing ICIAP 2005*, vol. 3617 of Lecture Notes in Computer Science, pp.50-57, Springer Berlin Heidelberg, 2005.

[2] W. Lee and S. J. Stolfo, "Data mining approaches for instrusion detection," *In Proceedings of the 7th USENIX Security Symposium*, vol. 7, pp. 6-6, Berkeley, CA, USA, 1998.

[3] C. Sinclair et al., "An application of machine learning to network instrution detection," *In Proceeding of the 15th Annual Computer Security Applications Conferences, ACSAC'99*, Washington, DC, USA, 1999.

[4] W. Li, "Using genetic algorithm for network intrusion detection," *In Proceedings of the US DoE Cybersecurity Conference*, Kansas City, KS, USA, 2004.

[5] S. Mukkamala, G. Janoski and A. Sung, "Intrusion detection using neural networks and support vector machines," *In Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN)*, vol. 2, pp. 1702-1707, 2002.

[6] J. Gomer and D. Dasgupta, "Evolving fuzzy classifiers for intrusion detection," *In Proceedings of*

the 2002 IEEE Workshop on Information Assurance West Point, NY, USA, 2002.

[7] A. Steven, S. Forrest and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of Computer Security*, vol. 6, no. 3, pp. 151-180, August 1998.

[8] J. Kim, J. Peter, U. Aickelin, J. Greensmith, G. Tedesco and J. Twycross, "Immune system approaches to intrusion detection," *Natureal Computing*, vol. 6, no. 4, pp. 413-466, December 2007.

[9] M. Zamani et al., "A DDoS-aware IDS model based on danger theory and mobile agents," *In Proceedings of the 2009 International Conferences on Computational Intelligence and Security*, vol. 1, pp. 516-520, 2009.

[10] M. Zamani et al., "A danger-based approach to intrsusion detection," *CoRR*, 2014.

[11] X. Glorot, A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks," *International Conference on Artificial Intelligence and Statistics*, 2011.

[12] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[13] A. L. Mass, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *In ICML*, vol. 30, 2013.

[14] D. Clevert, T. Unterthiner and S. HochreiterAcb, "Fast and Accuracy Deep Network Learning by Exponential Linear Units," *http:/arxiv.org/abs/1511.07289*, 2016.

[15] K. Jihyun, K. Jaehyun, L. T. T. Huong and K. Howon, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," *International Conference on Platform Technology and Service*, 2015.

Authors

Thi-Thu-Huong Le received her master degree from Hanoi University of Science and Technology (HUST), Vietnam, 2013. She had seven years' experience as a lecturer at Hung Yen University of Technology and Education (HYUTE), Vietnam. Currently, she is a Ph.D. candidate at Pusan National University (PNU), Korea. Her research interests include machine learning, deep learning, NILM, IDS.

Jihyun Kim received his master degree from Pusan National University (PNU), Korea, 2012. Currently, he is a Ph.D. candidate at Pusan National University (PNU), Korea. His research interests include machine learning, deep learning, NILM, IDS.

Howon Kim received his Ph.D. degree from POSTECH university, Korea, 1999. His employment experiment included research/team leader at ETRI from 1998 to 2008. Currently, he is a Professor at Pusan National University (PNU), Korea. His research interests include IoTs, Smart Grid Security, RFID/USN Security, PKC Cryptography, VLSI, and Embedded System Security